AF-Net: A Convolutional Neural Network Approach to Phase Detection Autofocus

Chi-Jui Ho, Chin-Cheng Chan, and Homer H. Chen, *Fellow, IEEE*

***Abstract*— It is important for an autofocus system to accurately and quickly find the in-focus lens position so that sharp images can be captured without human intervention. Phase detectors have been embedded in image sensors to improve the performance of autofocus; however, the phase shift estimation between the left and right phase images is sensitive to noise. In this paper, we propose a robust model based on convolutional neural network to address this issue. Our model includes four convolutional layers to extract feature maps from the phase images and a fully-connected network to determine the lens movement. The final lens position error of our model is five times smaller than that of a state-of-the-art statistical PDAF method. Furthermore, our model works consistently well for all initial lens positions. All these results verify the robustness of our model.**

***Index Terms*— Phase detection autofocus, supervised learning, focus profile, phase shift.**

I. Introduction

AUTOFOCUS is commonly needed for cameras to automatically capture sharp images [14], [27], [30]–[32]. The flowchart of a general autofocus scheme is shown in Fig. 1, where a certain focus measurement of the captured image is used to determine the lens movement. The decision of lens movement continues until an in-focus image is captured. A popular focus measurement technique used in smartphone cameras today involves the embedding of left and right phase detectors in the image sensor. The phase shift between the left and right phase images generated by the left and right phase detectors [1], [2], [16] represents the level of focus. If the lens is at the in-focus position, zero or nearly zero phase shift is resulted since the left and right phase images are identical. On the other hand, if the lens is at an out-of-focus position, a phase shift between the two phase images is resulted [2], [4]. An autofocus technique using such phase shift information is known as phase detection autofocus (PDAF).

Manuscript received January 7, 2019; revised June 23, 2019 and September 4, 2019; accepted October 1, 2019. Date of current version July 2, 2020. This work was supported in part by the Ministry of Science and Technology of Taiwan under Grant 106-2221-E-002-201-MY3. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Pavan Turaga. *(Corresponding author: Homer H. Chen.)*

C.-J. Ho and C.-C. Chan are with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: b04507009@ntu.edu.tw; b03901057@ntu.edu.tw).

H. H. Chen is with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, with the Graduate Institute of Communication Engineering, National Taiwan University, Taipei 10617, Taiwan, and also with the Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 10617, Taiwan (e-mail: homer@ntu.edu.tw).

This article has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TIP.2019.2947349

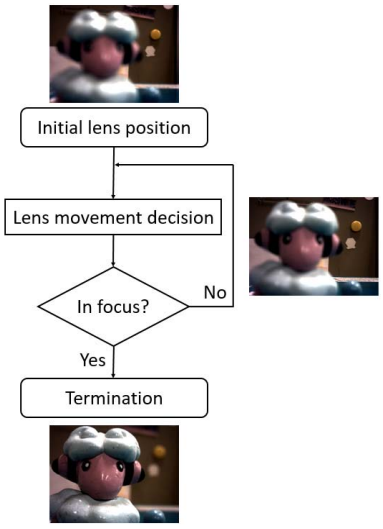

Fig. 1. A simplified flowchart of an autofocus system.

For PDAF, the sign of phase shift determines the direction of lens movement, and the magnitude of phase shift determines the lens travel distance. The phase shift is positive if the focal plane is in front of the object and negative if the focal plane is behind the object. Moreover, a large phase shift is resulted when the focal plane is far from the object, whereas a small phase shift is resulted when the focal plane is close to the object.

Both statistical [16] and reinforcement learning [24] approaches have been developed to determine the lens movement from phase shift. However, the phase shift estimation is seldom error-free. Sensor noise, image blur, and low-texture often affect the accuracy and robustness of phase shift estimation. Although the phase shift estimation error can be alleviated by applying a Gaussian filter to phase correlation [2], it is still difficult to obtain an accurate phase shift estimate when the lens is far from the in-focus lens position. Another notable problem of PDAF is that phase shift estimation error may accumulate in the lens movement decision process, particularly when the phase shift estimation process is performed separately [16], [24].

To reduce the impact of the phase shift estimation error on PDAF, we propose a model based on convolutional neural network (CNN) that directly determines the lens movement from the left and right phase images. CNN has been applied to estimate the disparity between stereo images [3], [11]–[13] or the optical flow of an image sequence [8]–[10]. However, the

left and right phase images in the context of this work are different from a pair of stereo images or two consecutive frames of a dynamic image sequence. Consider an in-focus object, the corresponding phase shift between the left and right phase images is zero. That is, the object is collocated in the two phase images. However, this is not the case for images captured from parallel stereo cameras unless the object is at a distance. In terms of image formation modeling, most stereo matching algorithms use the pinhole model because the input data are sharp images. This model is not applicable to autofocus because the input images are blurry unless the lens reaches the in-focus position. Therefore, a more realistic model such as the thin-lens model is required. In addition, the output of the CNN model for PDAF is a signed number representing the sign and magnitude of lens movement, not a flow map or a disparity map.

A specific CNN-based model is developed in this work for PDAF. The model uses four convolutional layers to extract feature maps from the left and right phase images. Then, it uses a fully-connected network to determine the lens movement from feature maps. The contributions of this paper are as follows:

- Our CNN model is robust to noisy phase shift and works equally well for all initial lens positions.
- Our CNN model is able to accurately estimate the distance to in-focus lens position from blurry images.
- The final lens position error of our model is much smaller than that of statistical PDAF method.
- This CNN-based approach is able to deal with small offset between the left and right phase detectors.

II. Related Work

In this section, we review previous work on PDAF and CNN-based models for optical flow, stereo matching, and autofocus.

A. Phase Detection Autofocus

The PDAF technique has been adopted for smartphone cameras to enhance the performance of autofocus because it is more accurate than the contrast detection autofocus (CDAF) technique when the image is blurry, which is normally the case at the beginning of an autofocus process [22]. The phase detectors are embedded in CMOS sensors using, for example, black masks [4] or dual photodiodes [23].

Example images for an illustration of phase shift are shown in Fig. 2. The phase shift between the phase images can be estimated in the image domain [1] or the frequency domain [2], [4]. To estimate the phase shift in the image domain, Wadhwa *et al.* [1] performed a 1D exhaustive search. It first computes the sum of squared differences (SSD) of various integral shifts between the left and right phase images. Then, the phase shift is the peak of the quadratic curve fitted on the SSD data. Phase correlation is commonly adopted for phase shift estimation in the frequency domain; however, it can easily fail for blurry or noisy images. To address this issue, Chan *et al.* [2] applied a Gaussian filter to the result of phase correlation, and Jang *et al.* [4] used difference

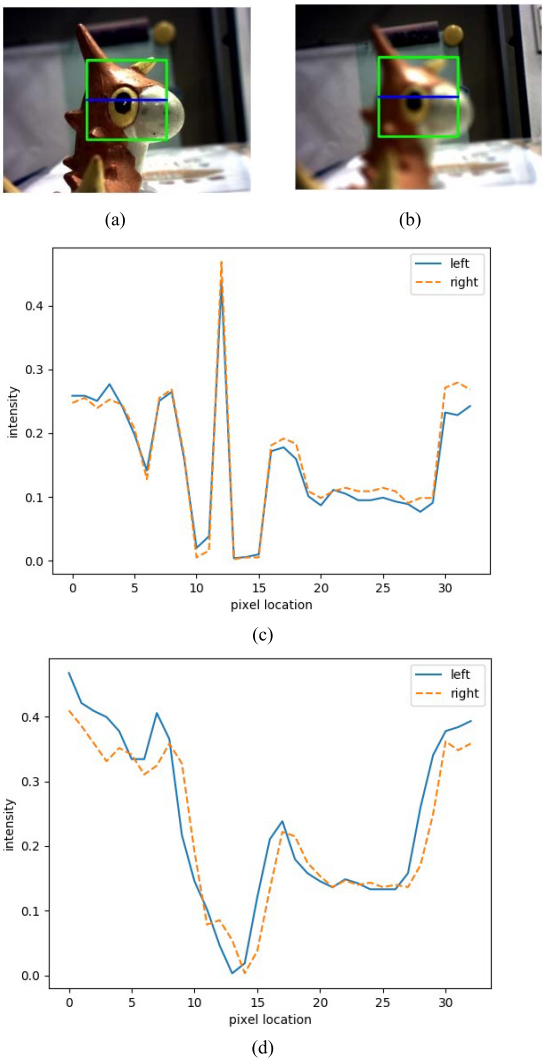

Fig. 2. Examples of (a) in-focus and (b) out-of-focus images. The center box represents the focus window, and the corresponding normalized pixel values along the middle row of the focus window of (a) and (b) are illustrated in (c) and (d), respectively.

of Gaussian to extract features from phase images before applying phase correlation to phase images.

A key element of PDAF algorithms is the characterization of the relation between phase shift and lens travel distance. At the camera calibration stage, one can incrementally move the lens across its motion range to determine the position of the focal plane. With the paraxial and thin-lens approximation of image formation [1], the object depth D is related to the phase shift s by

$$s = A\left(\frac{1}{z} - \frac{1}{D}\right), \tag{1}$$

where A is a constant and z is the distance from the focal plane to the lens. When the object is in focus, $z = D$. Accordingly, at the testing stage, the lens travel distance can be obtained by subtracting the current lens position from the corresponding focal plane position. However, Eq. (1) may not hold in the presence of phase shift estimation error. When the left and right phase detectors are not collocated, the offset between them may introduce phase shift estimation error.

Other factors such as the sparsity of phase detectors, sensor noise, and image blur may also cause phase shift estimation to drift. To address this issue, Chan and Chen [16] proposed a statistical method that first obtains the probability distribution of the optimal lens travel distance for a given phase shift at the calibration stage, and then uses it to determine the lens travel distance at the testing stage.

B. Convolutional Neural Network

Convolutional Neural Network (CNN) is a deep learning technique trained by backpropagation [21]. CNN works well for the estimation of stereo disparity [3], [11]–[13] or the optical flow between neighboring frames of a video [8]–[10]. In the area of stereo disparity estimation, Zbontar and LeCun [3] were among the first to propose a deep Siamese network (MC-CNN) for computing the similarity between stereo patches. In the area of optical flow estimation, Fischer *et al.* [9] pioneered the encoder-decoder architecture for computing the flow map. Many variants of these two models have been developed for further performance improvements, including a more accurate depth estimate for texture-less regions [13] and a smaller model size without affecting the overall performance [8].

CNN-based approach has been applied to autofocus for microscope images. For examples, Pitkäaho *et al.* [6], Yang *et al.* [7], and Wei and Roberts [26] applied CNN-based classifiers to predict the level of defocus. The data augmentation, which was applied to generate more defocused images for model training, was performed by either holographic reconstruction or synthetization. However, only ordinary images, not phase images, were considered. Besides, the simulated training data may not work well in practice. In our algorithm, we use real PDAF data for model training.

III. Phase Shift Estimation

The phase shift between the left and right phase images is commonly estimated by correlation. In this section, we describe the details of the phase shift estimation and the impact of phase shift error on the performance of PDAF.

A. Phase Shift Estimation

An ideal left and right phase image pair are related to each other by the phase shift Δx as follows:

$$r(x, y) = l(x + \Delta x, y), \tag{2}$$

where (x, y) denotes the pixel coordinates, $l(\cdot, \cdot)$ denotes the left phase image and $r(\cdot, \cdot)$ the right phase image. As an illustration, the values of pixels along the center row of the phase images within the focus window are shown in Figs. 2(c) and 2(d).

The phase shift between the left and right phase images can be obtained by phase correlation in, for example, the frequency domain. Denote the 2D Fourier Transform of the left phase image and right phase images by L and R, respectively. The phase shift is obtained by finding the peak of the correlation matrix $p(x, y)$,

$$p(x, y) = F^{-1}\left\{\frac{L \circ \bar{R}}{|L \circ R|}\right\}, \tag{3}$$

where $F^{-1}\{\cdot\}$ denotes the inverse 2D Fourier transform and the symbols "$\circ$" and "-" denote element-wise multiplication and complex conjugate, respectively. The x-coordinate of the peak is the phase shift between the left and right phase images.

However, the presence of sensor noise may generate multiple peaks in the correlation matrix. Chan et al. proposed to address the issue by smoothing the correlation curve $p(x, 0)$ with a Gaussian kernel $g(x)$ [2],

$$\mathrm{p}_f(x) = g(x) * p(x, 0), \tag{4}$$

where $\mathrm{p}_f(x)$ denotes the filtered correlation curve. The phase shift Δx is determined from the peak of the filtered correlation curve,

$$\Delta x = \arg\max_x \mathrm{p}_f(x). \tag{5}$$

Eq. (5) only gives rise to integral phase shift. Subpixel phase shift can be obtained by using the interpolation-based method proposed by Tian and Huhns [5].

If we plot the phase shift against the lens position, the resulting curve is called phase shift profile. The lens position corresponding to zero phase shift is the in-focus lens position. For an error-free phase shift profile, the phase shift is roughly proportional to the lens position in a finite region around the in-focus lens position. Beyond this region, the phase shift may saturate [2]. Also, the phase shift profile is not necessarily symmetric with respect to the in-focus lens position. Likewise, if we plot the image contrast against the lens position, a focus profile is resulted. In the ideal case, the lens position corresponding to the peak image contrast in the focus profile should correspond to a zero phase shift in the phase shift profile. Examples of phase shift profile and its corresponding focus profile are shown in Fig. 3, for which the phase shift is obtained by correlating the focus window (33×33 pixels) of the left and right phase images.

B. Effects of Phase Shift Error

Phase shift error may occur due to noisy image sensors, image blur, or lack of texture information. In practice, low-density phase detectors on the image sensor and the spatial offset between corresponding left and right phase detectors may cause erroneous phase shift estimation as well. The erroneous phase shift can cause an autofocus process to terminate prematurely or drag unnecessarily, resulting in either a blurry image or a long autofocus process.

Consider the noisy phase shift profile and its corresponding focus profile in Fig. 3(b). When the lens is far from the in-focus lens position, a large movement should be made so that the in-focus lens position can be quickly reached. However, the erroneous lens movement estimate in the presence of phase shift error results in a bumpy autofocus process. This can be prevented by scaling down the lens travel distance. The result is a graceful but slow autofocus process.

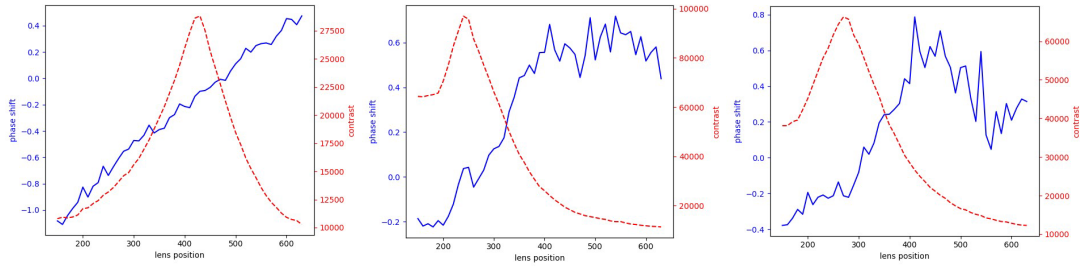

Fig. 3. Phase shift profile with (a) reasonable quality, (b) poor but manageable quality, and (c) the worst quality. The red (dash) curve is the corresponding focus profile, which expresses the image sharpness as a function of lens position.

TABLE I
CONFIGURATION OF THE PROPOSED AF-NET

Feature Extractor				
Name	kernel	strides	I/O channels	I/O size
conv0	5×5	2×2	2/64	$33 \times 33/16 \times 16$
conv1	3×3	1×1	64/128	$16 \times 16/16 \times 16$
conv2	3×3	2×2	128/128	$16 \times 16/8 \times 8$
conv3	3×3	2×2	128/256	$8 \times 8/4 \times 4$
Fully-Connected Network				
Name	I/O dimension		Activation function	
fc0	4096/256		ReLU	
fc1	256/64		ReLU	
fc2	64/1		linear	

Uniform PDAF performance for all possible initial lens positions in all cases is desirable. However, the presence of phase shift error can easily cause the phase shift profile to fluctuate or even change shape (see Fig. 3).

IV. MODEL TRAINING

Our approach avoids the noise sensitive operations of previous approaches and determines the lens movement directly from the phase images. The proposed model is called "AF-Net." We describe the details of this model in this section.

A. Model Design

The configuration of the model is shown in Table I. The AF-Net takes a pair of phase images as input and outputs a signed value. The sign represents the movement direction and the magnitude represents the travel distance for the lens. The first major component of the AF-Net is a feature extractor consisting of four convolutional layers. Every convolutional layer is followed by a Rectified Linear Unit (ReLU) [19] and a batch normalization [17]. We stack the two input phase images and feed them into the feature extractor to generate a feature volume. The image stacking is performed to allow for the application of 3D kernels in the first convolutional layer. These 3D kernels simultaneously extract the features of phase images and the displacement between them. Then, the feature volume is flattened into a feature vector. The second major component of the AF-Net is a three-layer fully connected network that estimates the optimal lens movement from the feature vector.

B. Data Collection

To train the CNN model, a large dataset is required. We use the PDAF platform shown in Fig. 4(a) to collect data from a large number of scenes. Two examples of the collected data are shown in Fig. 5. For all image data, we placed the focus window in the middle of the image. For each scene, we sweep the lens through its motion range at a constant step size to capture a sequence of images (a focal stack) of the scene. The motion range of the lens in our platform is 480, and the step size is 10. Therefore, the total number of images in each focal stack is 49.

The dataset includes both close shots and long shots to ensure data diversity. Close (long) shots refer to the case when the object is close to (far from) the camera. As shown in Fig. 6, when a close (long) shot is taken, the in-focus lens position is close to the far-end (near-end) of the lens motion range. The lens position measured with respect to the in-focus lens position is positive (negative) when the focal plane is in front of (behind) the object. In the data collection process, the lens is swept across its motion range at a constant step size; therefore, a large percentage of the initial lens positions would be negative (positive) for close (long) shots.

The size of the raw image is 1640×1232 pixels. The placement of phase detectors in our image sensor is shown in Fig. 4(b). The right phase detector is placed four pixels above the left phase detector. The left and right phase detector pair is repeated horizontally every 16 pixels and vertically every eight pixels. But the even and odd rows of phase detectors are placed with an offset of eight pixels, as shown in Fig. 4(b). The pixel values of the left (right) phase detectors are collected to form the left (right) phase image. An example of the left phase image structure is shown in Fig. 4(c). Since the odd and even phase detectors are not vertically aligned, we generate odd and even phase images separately. In other words, we generate two pairs of phase images from each raw image. Then, the average output of our model with odd and even a pair of phase images as inputs is the lens movement estimate. The size of odd and even phase images is 102×77 pixels, and the size of the focus window is 33×33 pixels.

C. Data Pre-Processing

We remove problematic data from model training. Problematic focal stacks are present in our dataset due to

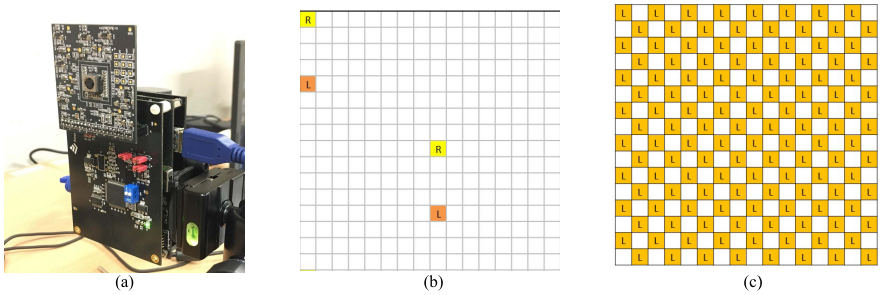

Fig. 4. Sensor pattern. (a) Our PDAF platform. (b) The pattern of left and right phase detectors of our PDAF platform. (c) Illustration of the assembled left phase image. Note that the even pixels and odd pixels are not vertically aligned.

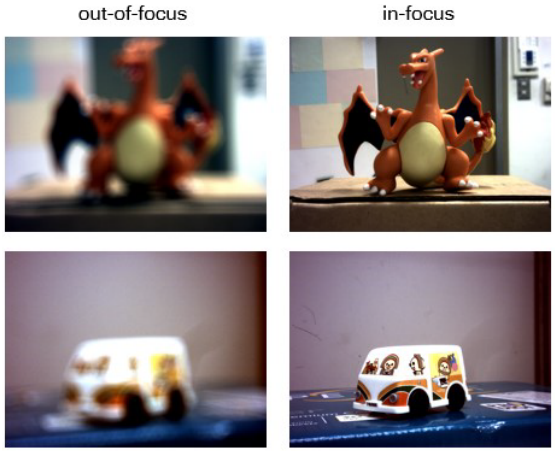

Fig. 5. Examples of in-focus images (right) and out-of-focus images (left) in our dataset.

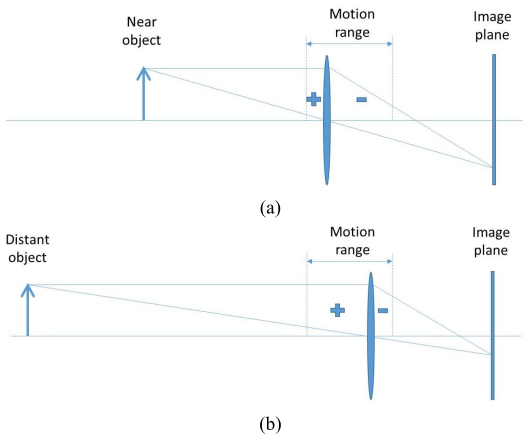

Fig. 6. Illustrations of positive and negative lens positions for (a) close shot and (b) long shot.

various human errors such as the accidental movement of the camera or lighting change. Objects or camera in our setup may be inadvertently moved during the course of data collection. A slight inadvertent object movement may cause a huge error to the phase shift data. The quality of a phase shift profile and hence the corresponding focal stack is judged by the extent of fluctuation in the phase shift profile.

The phase shift profile of each focal stack is fitted with a straight line. Then, the focal stacks with residual larger than a threshold are removed. In our experiment, the threshold is empirically set to 4. After removing problematic focal stacks, 822 focal stacks are left in our dataset. We partition these qualified focal stacks into three groups of size 660, 81, and 81 for training, validation, and testing, respectively.

The effectiveness of our CNN model depends in part on the quality of the training data. Besides the removal of problematic focal stacks, we need to find the in-focus lens position of each focal stack. This is achieved by finding the lens position corresponding to the sharpest image in each focal stack. In this work, the contrast $C(I)$ of an image I is measured by image gradient [15],

$$C(I) = \sum_{(x,y)\in I} \sqrt{G_x(x,y)^2 + G_y(x,y)^2}, \tag{6}$$

where

$$G_x(x,y) = 2I(x,y) - I(x-1,y) - I(x+1,y), \tag{7}$$

and

$$G_y(x,y) = 2I(x,y) - I(x,y-1) - I(x,y+1). \tag{8}$$

Note that the image gradient may fail to reflect the true in-focus lens position when imaging, for example, a point light source with an intensely bright light shining into the camera [28]. Therefore, we avoid collecting such images in our training dataset.

For the image with the highest contrast, the corresponding lens distance to the in-focus lens position is zero. For any other image in a focal stack, the corresponding lens distance to the in-focus lens position can be calculated by subtracting the in-focus lens position from the lens position corresponding to the image. This distance is taken as the ground truth. As an example, suppose the in-focus lens position of a focal stack captured by our imaging platform is 15. Then, the corresponding lens distance to the in-focus lens position of the images in the focal stack is $-15, -14 \ldots 0, 1, \ldots, 33$. The unit distance is the step size defined in Sec. IV.B.

In addition, we normalize the phase images to reduce the impact of lightness offset between the left and right phase images on the phase shift estimation. The normalized image $N(I)$ of a phase image I is obtained by

$$N(I) = \frac{I - \min(I)}{\max(I)}. \tag{9}$$

V. EXPERIMENTS

We compare the PDAF performance of four different models: AF-Net, statistical PDAF method [16], FlowNetC [9], and MC-CNN [3]. The latter two models were originally developed for optical flow and stereo disparity estimation. We modify and apply them to PDAF. The details of the modification are described in this section.

A. Experimental Setup

We start the lens of our platform shown in Fig. 4(a) at a number of initial distances for all models and measure their PDAF performance in terms of accuracy and speed. Each initial lens distance is measured with respect to the in-focus lens position, which is known *a priori*. A PDAF process is terminated when the lens is sufficiently near the estimated in-focus lens position in two consecutive lens movements or when a maximum number of lens movements is reached. In the experiments, the maximum number of lens movements is set to seven. Furthermore, the nearness threshold is set to $3\times$ stepsize, and the range of lens positions within the threshold is called near-focus region. A PDAF process is considered successful if the lens is within the near-focus region when the PDAF is terminated.

B. Metrics

Three metrics are used to measure the performance of each PDAF model: success rate, the number of lens movements, and the final lens position error. The success rate is the number of successful PDAF processes divided by the total number of PDAF processes in the test. It measures how successful a PDAF model is. The number of lens movements refers to the number of lens movement decisions required to complete a successful autofocus process. The smaller the number, the better the PDAF model is. The final lens position error refers to the absolute distance between the final lens position and the actual in-focus lens position. This metric helps differentiate models with a similar success rate.

C. Implementation

Our model is implemented using PyTorch, which is a library for the rapid implementation of machine learning models [20]. The model is trained using Adamax [18] as the optimizer, and mean-square-error as the loss function. The batch size is 128 and the parameters of the optimizer are $\beta_1 = 0.5, \beta_2 = 0.999$, and the learning rate is 0.001. We terminate the training process if the minimum loss of the model tested on the validation data does not decrease for 20 epochs. On average, the training process takes about 80 epochs.

D. Modification of FlowNetC and MC-CNN

Among the four PDAF models considered in the experiments, FlowNetC [9] and MC-CNN [3] require modification because the original outputs of these two models are flow map and matching cost, not lens movement.

For FlowNetC, the deconvolution layers are replaced by three fully-connected layers so that the network outputs a signed value representing lens movement. Moreover, the FlowNetC needs to determine the location of the correlation layer to optimize its performance. This is achieved by exhaustively testing each possible location of the correlation layer and recording the corresponding PDAF performance. The results show that placing the correlation layer between the first and the second convolutional layers yields the best PDAF performance.

For MC-CNN, we modify the training schedule. MC-CNN was originally developed to estimate the similarity (instead of the displacement) between two images. It cannot differentiate front-focus images from back-focus images. As a result, it can only be used to determine the magnitude of lens movement. We use phase correlation to determine the direction of lens movement. For fair comparisons, the modified FlowNetC and MC-CNN are retrained on our dataset.

E. Results and Discussions

Table II shows the performance of the four PDAF models tested on all test scenes with various initial lens positions. In average, the AF-Net has the highest success rate (95.98%), the smallest number of lens movements (2.07 movements), and the least final lens position error (1.094 stepsizes). Overall, the AF-Net performs consistently better than the other PDAF models across the range of initial lens positions. In cases such as over-exposure, the AF-Net may obtain a less than perfect result, but it seldom failed completely.

The results also show that the performance of AF-Net for positive and negative initial lens positions is not symmetric. Recall that the in-focus lens position is close to the far (near) end of lens motion range for close (long) shots, as shown in Fig. 6. Therefore, a large percentage of initial lens positions for close (long) shots is negative (positive). The effect of reflection, backlighting, and over-exposure on image quality and hence autofocus performance is more pronounced for close shots than for long shots. In other words, long shots have a higher success rate than close shots. Consequently, the algorithm performs better when the initial lens positions relative to the in-focus position is positive.

Furthermore, we compare the four PDAF models in terms of maximum final lens position error, maximum number of lens movement, and average run time per lens movement decision. The results are shown in Table III. The maximum final lens position error and the maximum number of lens movement for the AF-Net are 5 and 4, respectively, which are lower than those of the other models. In terms of average run time tested on a personal computer with an Intel i7-7700 CPU @ 3.60 GHz and an NVIDIA GTX 1080, the AF-Net takes 10.8 ms per lens movement decision, which is shorter than other models. As the hardware technology advances in the future, we believe the computational issue will be of less concern.

The performance of the FlowNetC is slightly worse than that of the AF-Net. Both methods combine features of the left and right phase images in the convolutional layers. However, the features are combined in different ways. The AF-Net

TABLE II
PERFORMANCE OF DIFFERENT PDAF MODELS

Model	Initial Lens Position Relative to the In-Focus Lens Position								
	-30			-20			-10		
	Success Rate (%)	Final lens position error	Number of lens movements	Success Rate (%)	Final lens position error	Number of lens movements	Success Rate (%)	Final lens position error	Number of lens movements
AF-Net	**90.48***	**1.381**	**2.26**	**94.83**	**1.207**	**2.20**	**95.06**	**1.185**	**2.03**
FlowNetC [9]	71.43	2.714	2.53	77.59	2.466	2.49	79.01	2.259	2.16
MC-CNN [3]	66.67	7.952	5.35	79.31	3.534	4.61	82.71	2.556	4.90
Statistical [16]	23.81	10.57	3.40	46.55	5.741	3.00	51.85	4.000	2.21
Model	Initial Lens Position Relative to the In-Focus Lens Position								
	10			20			30		
	Success Rate (%)	Final lens position error	Number of lens movements	Success Rate (%)	Final lens position error	Number of lens movements	Success Rate (%)	Final lens position error	Number of lens movements
AF-Net	**97.26**	**1.027**	**2.06**	**100.0**	**0.981**	**2.06**	**100.0**	**0.826**	**2.00**
FlowNetC [9]	83.56	2.164	2.16	88.89	1.796	2.40	91.30	1.565	2.38
MC-CNN [3]	79.45	2.356	4.69	88.89	2.148	4.89	91.30	2.217	5.00
Statistical [16]	63.01	3.342	2.21	57.41	3.426	2.54	43.48	3.261	2.60

*The highest performance is shown in boldface.

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT PDAF MODELS

Model	Max Final Lens Position Error	Max Number of Lens Movements	Average Runtime per Lens Movement (ms)
AF-Net	**5**	**4**	**10.8**
FlowNetC [9]	8	5	15.9
MC-CNN [3]	10	7	20.6
Statistical [16]	12	7	28.3

stacks and feeds the phase images to the convolutional layers, whereas the FlowNetC convolves feature maps extracted from the left and right phase images in the correlation layer. The main disadvantage of the latter is that image features may be lost after the convolution operation in the correlation layer. Since the AF-Net preserves more image features, it outperforms the FlowNetC in all tests.

The results also show that the MC-CNN is inferior to both FlowNetC and AF-Net. Although the MC-CNN also combines image features from the left and the right phase images, the features are extracted in separate convolutional layers. Consequently, the MC-CNN cannot estimate the phase shift between two phase images as accurately as the other models. Furthermore, erroneous estimate of the sign of phase shift results in incorrect lens travel direction and extra lens movements to reach the near-focus region. This is why the MC-CNN has worse performance than the AF-Net and the FlowNetC.

As shown in Table IV, the statistical method is unable to maintain consistent performance in the presence of noisy phase shift. To investigate it further, we classify the test focal stacks into two groups, clean and noisy, based on the smoothness of the corresponding phase shift profiles. Then we reorganize the PDAF performance of the AF-Net and the statistical method in the two groups accordingly. Table IV shows that the statistical method has a worse performance degradation than the AF-Net for noisy phase shift data. Its average success rate drops 14.49%, as opposed to 7.84% for the AF-Net.

To analyze the quality of video frames in the autofocus process, we test the four PDAF models with four different nearness thresholds. As shown in Table V, only the AF-Net performs constantly well in terms of success rate regardless of the nearness threshold. The chance of obtaining a sharp image is 79.56% even if a tight threshold is applied. We also calculate the average number of bounces of the lens movements in an autofocus process. Each time the lens changes its direction of movement, the bounce count is incremented. The bouncing leads to the alternation of image sharpness. Table VI shows that the AF-Net has the least bounces among the four PDAF models. Examples of video frames in the autofocus process using the AF-Net are shown in Fig. 7. In these examples, the autofocus processes start with an initial lens position that is away from the in-focus lens position. Therefore, the initial image is blurry. But a sharp image is obtained with only one lens movement. A comparison between the AF-Net with an autofocus method of smartphone camera can be found online [29].

To visualize how the AF-Net estimates the phase shift between phase images and determines the lens movement, we show in Fig. 8 the stimuli that yield the maximum response for sixteen filters randomly selected from the fourth convolutional layer of the AF-Net using the method proposed by Simonyan et al. [25]. From the stimuli, we can see that these filters are indeed capable of extracting features such as straight edges from phase images. By alternatively displaying the stimuli corresponding to the left and right

TABLE IV
PDAF PERFORMANCE TESTED ON CLEAN/NOISY DATA

Model	Initial Lens Position Relative to the In-Focus Position					
	-30			-20		
	Success Rate (%)	Final lens position error	Number of lens movements	Success Rate (%)	Final lens position error	Number of lens movements
AF-Net	**100.0/80.00**	**1.454/1.100**	**2.27/2.00**	**97.30/85.71**	**1.054/1.619**	**2.11/2.17**
Statistical [16]	36.36/10.00	5.636/16.00	3.00/5.00	51.35/38.10	4.054/8.714	2.48/4.24
Model	Initial Lens Position Relative to the In-Focus Position					
	-10			10		
	Success Rate (%)	Final lens position error	Number of lens movements	Success Rate (%)	Final lens position error	Number of lens movements
AF-Net	**98.04/90.00**	**1.117/1.300**	2.04/**2.00**	**100.0/92.59**	**0.913/1.222**	**2.07**/**2.04**
Statistical [16]	60.78/36.67	3.608/4.667	**2.00**/2.81	63.04/62.96	3.457/3.148	**2.07**/2.47
Model	Initial Lens Position Relative to the In-Focus Position					
	20			30		
	Success Rate (%)	Final lens position error	Number of lens movements	Success Rate (%)	Final lens position error	Number of lens movements
AF-Net	**100.0/100.0**	**0.869/1.235**	**2.08/2.00**	**100.0/100.0**	**0.786/0.889**	**2.00/2.00**
Statistical [16]	59.45/52.94	3.351/3.588	2.45/2.78	50.00/33.33	3.286/3.222	2.43/3.00

TABLE V
PDAF PERFORMANCE TESTED ON DIFFERENT NEARNESS THRESHOLDS

Model	Nearness Threshold					
	7			5		
	Success Rate (%)	Final lens position error	Number of lens movements	Success Rate (%)	Final lens position error	Number of lens movements
AF-Net	**100.0**	**1.145**	**2.00**	**99.20**	**1.124**	**2.00**
FlowNetC [9]	97.12	2.124	2.00	90.70	2.065	2.08
MC-CNN [3]	94.24	3.534	2.38	91.02	3.344	3.00
Statistical [16]	84.80	5.306	2.08	73.30	5.194	2.36
Model	Nearness Threshold					
	3			1		
	Success Rate (%)	Final lens position error	Number of lens movements	Success Rate (%)	Final lens position error	Number of lens movements
AF-Net	**95.98**	**1.094**	**2.07**	**79.56**	**1.061**	**2.70**
FlowNetC [9]	83.25	2.106	2.39	45.23	2.102	3.56
MC-CNN [3]	81.39	3.461	4.91	39.59	3.580	6.96
Statistical [16]	47.69	5.057	2.66	29.29	4.488	7.00

TABLE VI
AVERAGE BOUNCING TIME OF DIFFERENT PDAF MODELS

Model	Average Bouncing Lens Movements in Autofocus Process
AF-Net	**0.396**
FlowNetC [9]	0.554
MC-CNN [3]	2.900
Statistical [16]	0.792

phase images on a monitor (not shown), we can clearly see a displacement between the stimuli. These observations suggest that the responses of these filters are related to both image contrast and phase shift, indicating that the AF-Net uses these filters to calculate both the intensity of image features and the displacement between the left and right phase images. Collectively, all such information is used to estimate the lens movement. This explains the superior performance of the AF-Net.

VI. POSSIBLE EXTENSIONS

Recently, dual-pixel sensors [1] have been used in smartphones to perform PDAF. The configuration of such sensors is different from ours. The left and right pixels of dual-pixel sensors are collocated at the same pixel. Therefore, unlike our image sensor, there is no vertical offset between the left and

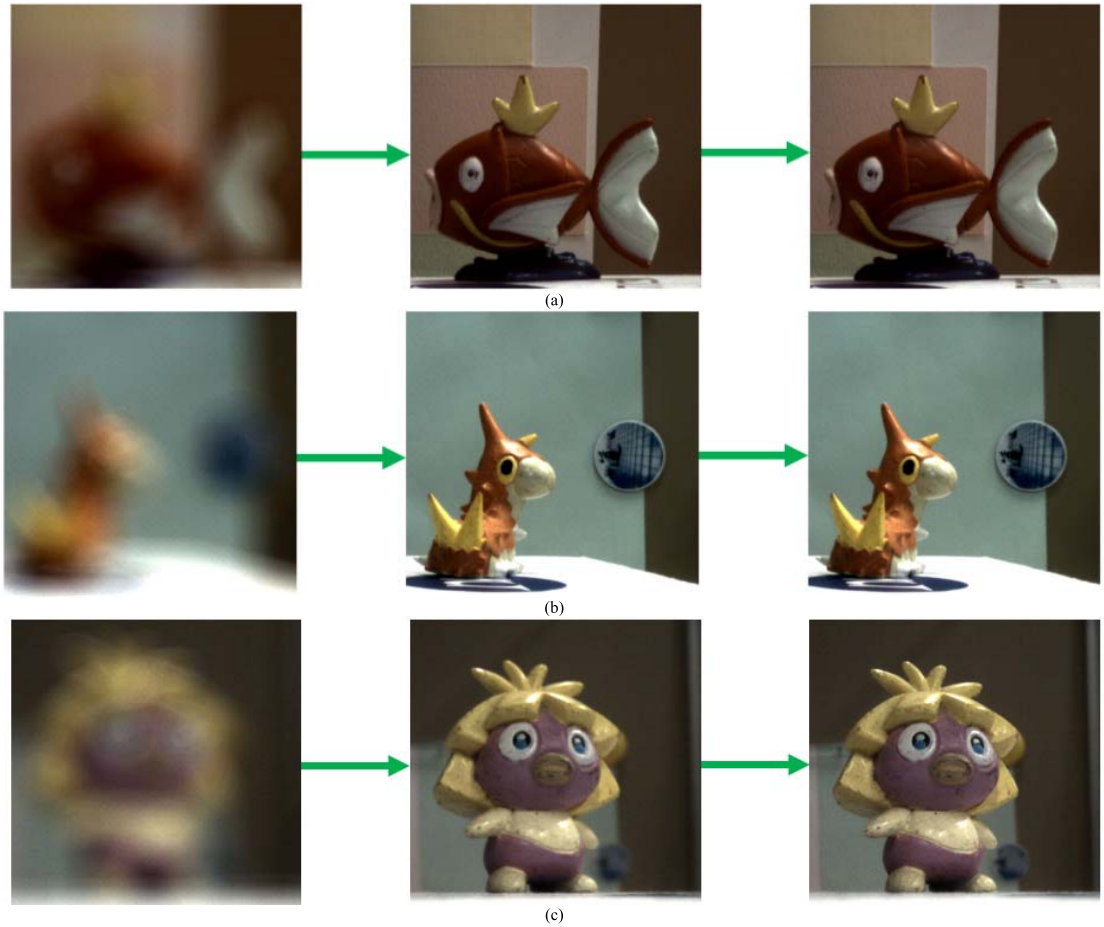

Fig. 7. Consecutive video frames of three scenes in the autofocus process of AF-Net. (a) Scene 1, (b) Scene 2, and (c) Scene 3.

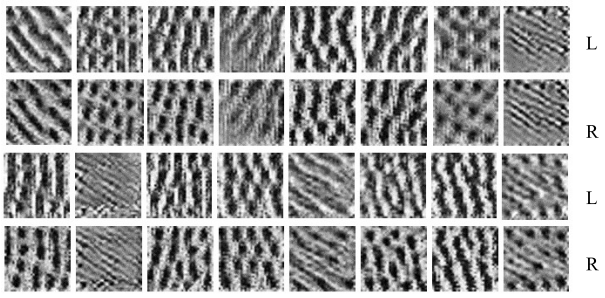

Fig. 8. Left and right stimuli that yield maximum response for sixteen randomly selected filters in the fourth convolutional layer of the AF-Net.

right phase images. Naturally, the resulting phase shift data are cleaner. It is straightforward to train our AF-Net for such sensors, and we believe better performance can be achieved because the data are less noisy.

The proposed AF-Net for the autofocus of still cameras can be extended to video cameras. The complexity of such continuous autofocus is higher than that of still autofocus described in this paper. The former needs to deal with dynamic scenes, whereas the latter only has to consider still scenes. The AF-Net can be directly applied to continuous autofocus when the focal planes for neighboring frames are close to each other. According to the results shown in Tables II and V, we believe the AF-Net is able to go from the near-focus lens position to the in-focus lens position in one lens movement for such cases.

Our PDAF algorithm can be extended to the case where the focus window is constantly adjusted to follow a moving object. This is useful for object tracking. We believe it can be achieved by integrating object detection with our autofocus technique. It involves moving the focus window according to the displacement of the target object.

VII. Conclusion

Maintaining consistent PDAF performance in the presence of noisy phase shift data is a challenging task. This paper has described a CNN-based PDAF model called AF-Net to address this issue. The AF-Net has superior performance in terms of accuracy, robustness, and speed. In 95% of the test cases performed in our experiments, it reaches the in-focus lens position in two lens movements on average and with final lens position error less than one stepsize.

References

[1] N. Wadhwa *et al.*, "Synthetic depth-of-field with a single-camera mobile phone," *ACM Trans. Graph.*, vol. 37, no. 4, Aug. 2018, Art. no. 64.

[2] C.-C. Chan, S.-K. Huang, and H. H. Chen, "Enhancement of phase detection for autofocus," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2017, pp. 41–45.
[3] J. Žbontar and Y. LeCun, "Computing the stereo matching cost with a convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1592–1599.
[4] J. Jang, Y. Yoo, J. Kim, and J. Paik, "Sensor-based auto-focusing system using multi-scale feature extraction and phase correlation matching," *Sensors*, vol. 16, no. 3, pp. 5747–5762, 2015.
[5] Q. Tian and M. N. Huhns, "Algorithms for subpixel registration," *Comput. Vis. Graph. Image Process.*, vol. 35, no. 2, pp. 220–233, Aug. 1986.
[6] T. Pitkäaho, A. Manninen, and T. J. Naughton, "Performance of autofocus capability of deep convolutional neural networks in digital holographic microscopy," in *Proc. Digit. Hologr. Three-Dimensional Imag.*, 2017, p. W2A-5.
[7] S. J. Yang *et al.*, "Assessing microscope image focus quality with deep learning," *BMC Bioinf.*, vol. 19, no. 1, 2018, Art. no. 77.
[8] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2018, pp. 8934–8943.
[9] P. Fischer *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2758–2766.
[10] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1647–1655.
[11] N. Mayer *et al.*, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2016, pp. 4040–4048.
[12] J. Pang, W. Sun, J. S. J Ren, C. Yang, and Q. Yan, "Cascade residual learning: A two-stage convolutional neural network for stereo matching," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, Jul. 2017, pp. 887–895.
[13] J.-R. Chang and Y.-S. Chen, "Pyramid stereo matching network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2018, pp. 5410–5418.
[14] A. Abuolaim, A. Punnappurath, and M. S. Brown, "Revisiting autofocus for smartphone cameras," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 523–537.
[15] L. C. Chiu and C. S. Fuh, "An efficient auto focus method for digital still camera based on focus value curve prediction model," *J. Inf. Sci. Eng.*, vol. 26, no. 4, pp. 1261–1272, Jul. 2010.
[16] C.-C. Chan and H. H. Chen, "Improving the reliability of phase detection autofocus," in *Proc. IS&T Electron. Imag.*, Jan 2018, pp. 241-1–241-5.
[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1–41.
[19] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 807–814.
[20] A. Paszke *et al.*, "Automatic differentiation in pytorch," in *Proc. NIPS Autodif Workshop*, 2017, pp. 1–4.
[21] Y. LeCun *et al.*, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.
[22] M. Subbarao, T.-S. Choi, and A. Nikzad, "Focusing techniques," *Opt. Eng.*, vol. 32, no. 11, pp. 2824–2836, Nov. 1993.
[23] M. Hamada, "Imaging device including phase detection pixels arranged to perform capturing and to detect phase difference," U.S. Patent 9 197 807, Nov. 24, 2015.
[24] C.-C. Chan and H. H. Chen, "Autofocus by deep reinforcement learning," in *Proc. Electron. Imag.*, 2019, pp. 1–6.
[25] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Proc. Int. Conf. Learn. Represent. Workshop*, 2014, pp. 1–8.
[26] L. Wei and E. Roberts, "Neural network control of focal position during time-lapse microscopy of cells," *Sci. Rep.*, vol. 8, May 2018, Art. no. 7313.
[27] R. Chen and P. van Beek, "Improving the accuracy and low-light performance of contrast-based autofocus using supervised machine learning," *Pattern Recognit. Lett.*, vol. 56, pp. 30–37, Apr. 2015.
[28] S.-K. Huang, D.-C. Tsai, and H. H. Chen, "Overcoming the blooming effect on autofocus by fringe detection," *Proc. SPIE*, vol. 9404, Feb. 2015, Art. no. 94040M.
[29] C. J. Ho. (2019). *Autofocus by Artificial Intelligence*. [Online]. Available: https://www.youtube.com/watch?v=ApXMDT774aA
[30] D.-C. Tsai and H. H. Chen, "Focus profile modeling," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 818–828, Feb. 2016.
[31] D.-C. Tsai, Z.-M. Tsai, and H. H. Chen, "A simulation model for continuous autofocus design," *IEEE Trans. Consum. Electron.*, vol. 59, no. 4, pp. 731–737, Nov. 2013.
[32] D.-C. Tsai and H. H. Chen, "Reciprocal focus profile," *IEEE Trans. Image Process.*, vol. 21, no. 2, pp. 459–468, Feb. 2012.

Chi-Jui Ho was born in Taipei, Taiwan, in 1996. He received the bachelor's degree in electrical engineering from National Taiwan University in 2019. He was a Summer Intern with MediaTek, Hsinchu, Taiwan, in 2018. His research interests include image processing and machine learning. His research topics include autofocus for smartphone cameras and deep learning analysis for biomedical images.

Chin-Cheng Chan received the bachelor's degree in electrical engineering from National Taiwan University in 2018. He was a Summer Intern with MediaTek, Taiwan, in 2016. His research interest includes image processing with a focus on computational imaging. His research topics include robust principal component analysis, autofocus for smartphone cameras, and deep learning for medical image analysis.

Homer H. Chen (S'83–M'86–SM'01–F'03) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign.

His professional career has spanned industry and academia. Since August 2003, he has been with the College of Electrical Engineering and Computer Science, National Taiwan University, where he is a Distinguished Professor. Prior to that, he held various research and development management and engineering positions at U.S. companies over a period of 17 years, including AT&T Bell Labs, Rockwell Science Center, iVast, and Digital Island. He was a U.S. delegate for ISO and ITU standards committees and contributed to the development of many new interactive multimedia technologies that are now a part of the MPEG-4 and JPEG-2000 standards. His professional interests lie in broad areas of multimedia signal processing and communications.

Dr. Chen was a Distinguished Lecturer of the IEEE Circuits and Systems Society from 2012 to 2013. He served on the IEEE Signal Processing Society Fourier Award Committee and the Fellow Reference Committee from 2015 to 2017. He serves on the IEEE Signal Processing Society Awards Board and the Senior Editorial Board of the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING. He is a General Chair of the 2019 IEEE International Conference on Image Processing. He was an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY from 2004 to 2010, the IEEE TRANSACTIONS ON IMAGE PROCESSING from 1992 to 1994, and *Pattern Recognition* from 1989 to 1999. He served as a Guest Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 1999, the IEEE TRANSACTIONS ON MULTIMEDIA in 2011, the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING in 2014, and *Multimedia Tools and Applications* (Springer) in 2015.