

RESEARCH ARTICLE

Depth-of-Field Region Detection and Recognition From a Single Image Using Adaptively Sampled Learning Representation

JONG-HYUN KIM¹ AND YOUNGBIN KIM², (Member, IEEE)

¹College of Software and Convergence (Department of Design Technology), Inha University, Michuhol-gu, Incheon 22212, South Korea

²Graduate School of Advanced Imaging Science, Multimedia and Film, Chung-Ang University, Seoul 06974, South Korea

Corresponding author: Youngbin Kim (ybkim85@cau.ac.kr)

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2022R1F1A1063180 (Contribution Rate: 60%); in part by the Basic Science Research Program through NRF funded by the Ministry of Education under Grant NRF-2022R1C1C1008534 (Contribution Rate: 20%); and in part by the Institute for Information and Communications Technology Planning and Evaluation (IITP) through Korean Government (MSIT) (Artificial Intelligence Graduate School Program, Chung-Ang University) under Grant 2021-0-01341 (Contribution Rate: 20%).

ABSTRACT This study describes a network and its application methods for efficient detection and recognition of the depth-of-field(DoF) region blurred in the image by focusing and defocusing the camera. This approach uses a cross-correlation filter based on RGB color channels to efficiently extract DoF regions in images and construct a dataset for training in the convolutional neural network. A data pair corresponding to the image-DoF weight map is set using the data. The training data are from a DoF weight map extracted based on an image and cross-correlation filter. The loss function is modeled using the result of applying Gaussian derivatives of the image to improve the convergence rate efficiently in the network training phase. The DoF weight map obtained as a test result and proposed in this paper reliably extracted the DoF region in the input image. In addition, this study experimentally demonstrates that the proposed method can be used in various applications, such as non-photorealistic rendering, viewpoint tracking, object detection and recognition, optical character recognition, and adaptive sampling, that employ the user regions of interest.

INDEX TERMS Depth of field, object detection, object recognition, quadtree, adaptive sampling, non-photorealistic rendering, viewport tracking, optical character recognition.

I. INTRODUCTION

One of the most commonly addressed tasks is depth estimation from an image, an analysis of the geometric relationships in a three-dimensional(3D) scene or a scene that exists implicitly in an image. This analysis of the relationship between an object and environment improves object recognition accuracy and applies to applications, such as 3D modeling [1], physically based modeling [2], self-driving vehicles [3], video representation [4], and robotics [5]. Several stereo-image-based techniques have been proposed as tools for depth map estimation. When the stereo-image-based technique is applied, blurring may occur

in the depth-of-field(DoF) region caused by the difference between focusing and defocusing. Another kind of blurring is motion blur caused by the speed of the content in the video. The DoF is one of the characteristics resulting from focusing the camera. This study proposes a framework to detect the DoF region contained in images efficiently through artificial neural networks.

In such processes as content analysis, object detection, and the user region of interest(ROI) calculation using image or video data, the DoF of an image is an important consideration [6], [7]. There is insufficient information to predict and approximate the focusing area from an image containing DoF; thus, it is difficult to identify the DoF regions within the image. To solve this problem, many researchers have attempted to analyze the geometric relationship by

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaogang Jin.



FIGURE 1. This depth-of-field(DoF) image is the result of using the proposed neural network system.

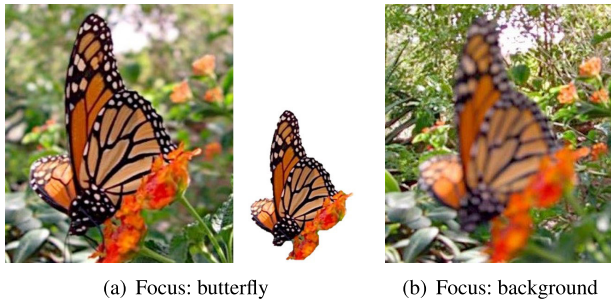


FIGURE 2. Results when the focusing position of the depth of field(DoF) is different (inset image in (a): focused object).

calculating the scene depth in an image [8]. However, most such methods were applied only to pose determination and object recognition and were unsuitable for identifying a specific ROI contained in a single image. The conventional DoF, unlike the depth of content, is a technique that emphasizes the part the users aim to focus on; thus, the DoF addressed in studies targeting depth estimation is different from that in this study. The difference between focusing and defocusing in an image is displayed by the difference between the clarity and blurring of colors, expanding when the colors in the image space are subjected to Gaussian derivatives. This study proposes a new neural network-based model using these characteristics. Fig. 1 visualizes the DoF regions extracted through our method, and in this paper, we propose an efficient method to identify the focusing and defocusing regions represented by DoF using a single image.

A. PROBLEM STATEMENT

In general, the DoF of an image influences object detection and recognition, rendering, ROIs, and viewpoint tracking. In image processing, although users may use the DoF in image editing by manually designating an area to which the DoF is applied, the DoF in an image is one of the common characteristics indicating the user ROI. Even for the same scene, the DoF region may be expressed differently depending on where to focus, which also affects the interpretation of an image.

Fig. 2 depicts the difference in the image according to the change of the focused object. The interpretation image, as mentioned, may depend on the focusing position of the DoF. In Fig. 2a and 2b, the images are interpreted differently due to the focus on the butterfly and background,



(a) Accuracy: 1.0, 0.998, 0.999, and (b) Accuracy: 0.998, 1.0, 0.909, 0.999, recognition: person, person, 0.999, and 0.999, recognition: person, person, person, cell phone, person, person

FIGURE 3. Image detection and recognition results tested on images with the depth of field(DoF) applied (object accuracy left to right).

respectively, though they are the same scenes. As such, the DoF characteristics can affect various applications.

As mentioned, the object interpretation varies depending on the focusing position of the image, and the tasks of image detection and recognition are directly related to this problem. Fig. 3 illustrates the results using YOLO [40] for image detection and recognition on a photo of four children. Although Fig. 3a and 3b have different focusing positions and ROIs, the people were successfully detected and recognized in both images.

In Fig. 3, all four children are recognized as people, and almost no difference in accuracy exists, leading to a different result from the user’s intention in the focus, which is also true in Fig. 3. The most important characteristic in the difference between the data in Fig. 3 and 3 is the viewpoint change. Although the viewpoint moves from left to right, it is challenging to extract such movement when the DoF information is unavailable.

It is not absolute that object ‘A’ will be recognized as object ‘B’ according to the DoF. However, the main point we wanted to convey is that in the same scene, various contents, or different objects, can exist, and if we assume that one object is being focused by the DoF, the recognition result can vary depending on the focused object. Depending on the strength of the DoF, it may or may not be possible to recognize it as a person. In the case of Fig. 3, most of the people were recognized as such, but the strength of the DoF could lead to incorrect recognition results. The example shown in Fig. 3 demonstrates not just simple recognition of a person, but rather the ability to determine which person to focus on. If there were a mixture of humans and animals in the scene, the DoF could lead to more clear recognition of the animal rather than the person. In conclusion, we do not believe that the two sentences are contradictory.

If the DoF is not considered in non-photorealistic rendering(NPR), the problem of a serious flattening of the object color occurs in the process of exaggeration into cartoon style, and, in optical character recognition(OCR), text recognition accuracy decreases (Fig. 4). As depicted in Fig. 4a, applying the NPR technique without considering the DoF causes color over-simplification in the focused monkey face. Fig. 4b, presenting the result of the OCR test, reveals that the text

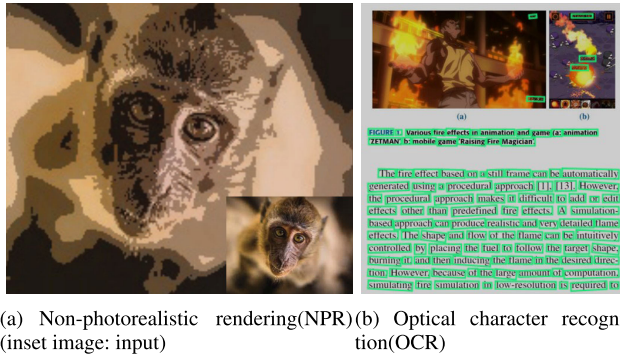


FIGURE 4. Results applied in various applications.

and the image caption with included text are recognized. Although the DoF may be considered in terms of reading and understanding the text, conventional approaches in the application of the DoF assume that all images are clean and do not distinguish between focusing and defocusing. This study proposes a method to detect the DoF region efficiently through a neural network and demonstrates the efficiency and utility of the proposed method by applying it through experiments to the mentioned applications.

B. RELATED WORK

The reconstruction of the depth map from an image is a classic topic in computer vision. This section explains a method to calculate the depth value from a single image and an approach to analyze the depth value using lens blur effects.

C. SINGLE-IMAGE DEPTH ESTIMATION

The convolutional neural network (CNN) has been used in many studies to train a network using a depth map obtained from the depth camera and predict a depth map of a single image based on it [9], [10], [11]. Due to limitations in terms of the range, image quality, portability, and resolution, the depth camera sometimes works only in certain scenes, such as indoor spaces [2], streetscapes [12], and landmarks [13]. Recently, aperture stacks have been used for depth value prediction and supervision, and their efficiency has been demonstrated in indoor spaces and flower images [14]. Approaches to supervise the depth values may cause depth ambiguities in some scenes, preventing the production of high-quality images.

D. MULTIPLE-IMAGE DEPTH ESTIMATION

The most common method to obtain depth information is using correlations obtained from multiple images, and attempts to implement a shallow DoF effect through this method have been made in various disciplines. These include such approaches as calculating the depth map from the camera focus on a mobile device [15], using a stereo camera pair [16], using a baseline [17], [18], and using data from multiple frames of an image [19]. These methods implement the shallow DoF effect using the focused image and the depth value calculated in the preprocessing step.

E. LENS BLUR EFFECTS

Lens blur effects applied to an RGB-D image allow the accurate representation of defocused regions hidden behind objects. The conventional methods to approximate lens blur generally distinguish object space and image space. Although the object space method effectively represents shallow DoF images by ray-tracing and real camera modeling [20], [21], it requires a long calculation time because complex 3D spatial information is needed [22], [23], [24]. In the image space method, shallow DoF effects are implemented through gathering and spreading process in a single image [22], [23], [24]. Mobile devices, such as the iPhone and Google Pixel 2, have Portrait Mode, a built-in application that simulates DoF images. This method approximates the depth map based on multiple views and dual pixels. Conventional methods have attempted to extract only the depth value within the image, and there has been no attempt to extract the DoF. In most cases, clean images are edited to implement focusing and defocusing effects. Although various methods have been applied to evaluate or calculate depth value, it is challenging to employ these methods to extract the precise DoF regions because the DoF and depth value have different characteristics.

F. IMAGE FILTERING AND CORRELATION

In this paper, as cross-correlation filters are used, image filtering and correlation techniques are important. Elad and Michael proposed a Bilateral filter that is widely used and incorporates methods for noise removal, anisotropic diffusion, and adaptive approaches based on weighted least squares and robust estimation [41]. There are also studies that have improved image fusion algorithms using these filtering and intensity matching techniques [42]. In image filtering, cross-correlation analysis and canonical correlation analysis are probabilistic analysis techniques that determine the linear relationship between two datasets [43]. These approaches are used in various fields such as image denoising, smoothing, and deblurring [44], [45]. Moreover, there are studies that use PCA (Principal Component Analysis) to detect historical invariant features from images [46]. The orientation extracted from PCA is also frequently used to determine the position and direction of brushes in the field of NPR (Non-Photorealistic Rendering) or painterly rendering [47], [48].

II. PROPOSED FRAMEWORK

In this section, the proposed method is presented in the following sequence: 1) extracting depth-of-field regions from images to build the training dataset, and 2) designing an artificial neural network for depth-of-field learning.

A. DEPTH-OF-FIELD REGION EXTRACTION FROM IMAGES FOR THE TRAINING DATASET

This subsection explains a method to build a dataset for the training phase. In general image super-resolution, a low-resolution image is generated by downsampling a

high-resolution image, and training is performed using the loss between these two images. A more specialized dataset is required to extract the DoF region, and a method for constructing such a dataset is described below.

In this paper, the cross-correlation filter \mathbf{G} calculates the DoF region of the image. This filter measures the association level between two consecutive data and is used in various fields, such as image processing and computer vision. Unlike the general depth field, DoF has the feature that it includes the user's RoI, and in this paper, we propose an efficient method to identify this feature. We use a cross-correlation filter to distinguish features that appear blurred in the defocusing region and those that appear sharp in the focusing region. To efficiently learn the DoF regions, we use Gaussian derivatives based on features that are close to 0 in defocusing and close to 1 in focusing.

DoF refers to the range in front and behind that is perceived as in focus, either by the camera or the user. This phenomenon is also reflected in object detection or recognition, which is of interest to the user. As shown in Fig. 2, it is the same scene, but depending on the object of focus, it could be a butterfly or the background could be recognized. In this paper, we experimentally discovered that the characteristics of DoF are consistently represented in space, and we used cross-correlation filters to efficiently identify them. We aim to efficiently detect the DoF area by identifying the changes that occur between the original image and the filtered image and to learn it through a neural network. In this paper, we used cross-correlation filters to calculate the spatially consistent clean/blurring characteristics of DoF. The filter is defined as follows (see Eq. 1)

$$\mathbf{G}(x, y) = \mathbf{H} \otimes \mathbf{F} = \sum_{u=-1}^1 \sum_{v=-1}^1 \mathbf{H}(u, v) \mathbf{F}(x+u, y+v), \quad (1)$$

where \mathbf{H} , called a filter, kernel, or mask, is the weight of each adjacent pixel, and \mathbf{F} is the color of adjacent pixels. The mask is modeled in various forms depending on the field to which it is applied. In this study, a Gaussian-type filtering technique is used (see Eq. 2):

$$\mathbf{H}(x, y) = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \approx \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{\sigma^2}}. \quad (2)$$

where σ is the variance. This study made the following assumptions before anisotropically estimating the DoF region of the image:

- 1) In the DoF region, the color around the focused region is gradually blurred (see Fig. 5a).
- 2) The area blurred by defocusing and the clear area are distinguished (see Fig. 5b).

Using the two mentioned features, an anisotropic DoF-weighted map \mathbf{D}^* was calculated based on the difference in RGB color between the original image I_0^{rgb} and its Gaussian derivatives image I_1^{rgb} (see Eq. 3). The final output, \mathbf{D}^* ,

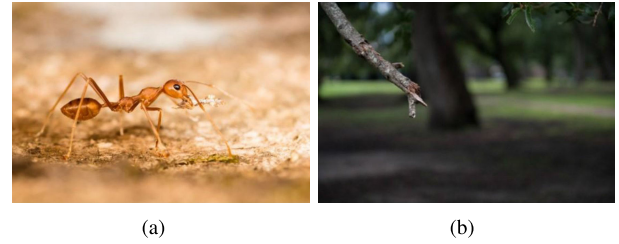


FIGURE 5. Focusing and defocusing regions in photography.

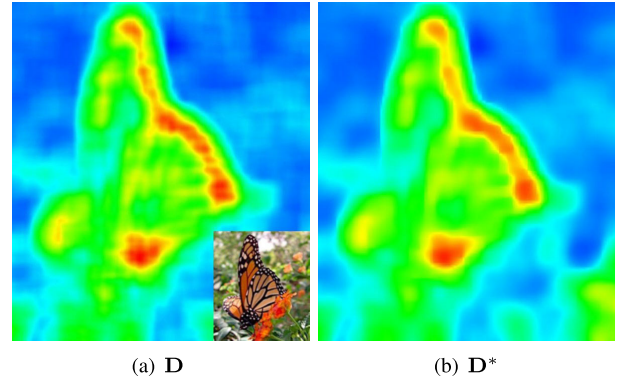


FIGURE 6. Weight maps calculated using the depth-of-field (DoF) region. Blur effects are stronger from blue to red (inset image: input).

is obtained by refining the DoF weight map anisotropically to account for potential noise that may arise if only \mathbf{D} is used.

To minimize the noise in the image calculated using the cross-correlation filter and extract the focusing map features anisotropically, we calculate \mathbf{D}^* using \mathbf{D} as follows:

$$\begin{aligned} \mathbf{D}^* &= \frac{\partial \mathbf{D}}{\partial t} = \text{div}(c(x, y, t) \nabla \mathbf{D}) \\ &= \nabla c \cdot \nabla \mathbf{D} + c(x, y, t) \Delta \mathbf{D} \end{aligned} \quad (3)$$

$$c(\|\nabla \mathbf{D}\|) = \frac{1}{1 + \left(\frac{\|\nabla \mathbf{D}\|}{K}\right)^2}. \quad (4)$$

where the constant K controls the sensitivity to edges, set to 2 in this paper.

Equation (3) is the anisotropic DoF filter proposed in this paper, where div , ∇ , and Δ represent the divergence, gradient, and Laplacian operators, respectively. In Eq. 4, c is the diffusion coefficient, and the cross-correlation filter \mathbf{D} is calculated using Eq. 5:

$$\mathbf{D}(x, y) = \frac{\sum_{u=-m}^m \sum_{v=-m}^m \mathbf{H}_m(u, v) \mathbf{O}(x+u, y+v)}{m^2} \quad (5)$$

$$\mathbf{O}(x, y) = \left\| I_0^{rgb}(x, y) - I_1^{rgb}(x, y) \right\|, \quad (6)$$

where \mathbf{H}_m and m represent the mask and its size, respectively, which is set to 15 in this study. In addition, \mathbf{O} represents the difference in color between the blurred and clean images. Moreover, I_0^{rgb} and I_1^{rgb} are RGB colors obtained from the original and blurred images, and the Gaussian smoothing technique is used as a smoothing filter.

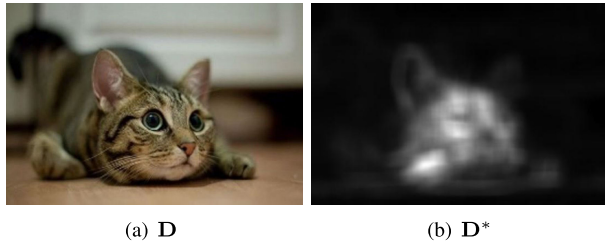


FIGURE 7. Depth-of-field(DoF) weight map \mathbf{D}^* calculated using the DoF region (white: focusing, black: defocusing).

In Eq. 3, c is an edge-stopping function that reduces or stops diffusion when the gradient value is high, considering it as an edge. In other words, the output of $c(x, y, t)$ is 0 when $\|\nabla \mathbf{D}\|$ is infinite, reducing the rate of diffusion, and it is 1 when $\|\nabla \mathbf{D}\|$ is 0, increasing the rate of diffusion. Generally, t is a Gaussian kernel in this process. As presented in the equation, the weight of the DoF region is calculated using the norm of the 3D vector, which is an expression of the RGB channel value. In Fig. 6, the resulting images corresponding to \mathbf{D} and \mathbf{D}^* were obtained through the above equations, and the weight of the defocused region in the input image is expressed well.

Anisotropic diffusion filtering is based on the scale-space theory and is established and used by applying scale-space filtering, such as $\mathbf{D}(x, y, t) = \mathbf{D}_0(x, y) \times \mathbf{G}(x, y, t)$. When $t > 0$, the output image is expressed as a blurrier image of the input image, and the larger t is, the more strongly it is expressed. $c(x, y, t)$ controls the rate of diffusion and is usually selected as a function of image gradation to preserve the edge of the image. Pietro Perona and Jitendra Malik pioneered the idea of anisotropic diffusion in 1990 and proposed two functions for the diffusion coefficient, and one of these functions is Eq. 4 used in this paper.

Fig. 7 depicts the DoF weight map \mathbf{D}^* obtained from the input image. The weight of the focused region using the DoF is expressed well, and a dataset for network training was constructed.

B. TRAINING DEPTH-OF-FIELD WEIGHT MAP WITH THE CONVOLUTIONAL NEURAL NETWORK

Using the described method, the input image with RGB color channels $\{\delta^1, \delta^2, \dots\}$ and DoF weight map image $\{\mathbf{D}^{*1}, \mathbf{D}^{*2}, \dots\}$ are generated. Before being input into the training network, each image is divided into patches. After the training data are prepared, the target is to determine the mapping function $f(x)$ to minimize the loss between the predicted value δ_s and the ground truth \mathbf{D}^* .

The goal of this equation is to find the function (model) f that best approximates the input image x into a DoF weight image. The desired image, or the approximated DoF weight map image, is $\delta_s = f(x)$, and the objective is to minimize the difference between this desired image and the actual ground truth DoF weight map image, \mathbf{D}^* . Specifically, the mean squared error (MSE) loss function is used to minimize



this difference. The objective function for this process is the mean squared error between the predicted and ground-truth images. The target is to train a model f that predicts the value of $\delta_s = f(x)$ and minimize the mean squared error for the training data L (see Eq. 7):

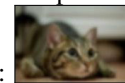

$$L = \frac{1}{2} \|\mathbf{D}^* - f(x)\|^2. \tag{7}$$

This study uses a method based on the super-resolution CNN to implement the loss function L in a simple way [25]. The application of the proposed method to the super-resolution CNN method requires calculating the weight parameter θ to convert the DoF weight map of the input image

into its Gaussian derivative image $\underbrace{\delta - \mathbf{D}^*}_{step1} - \left(\underbrace{\delta^\dagger - \mathbf{D}^\dagger}_{step2}, \theta \right)$,

where δ and \mathbf{D}^* are input image and its DoF weight map,

respectively (δ : , \mathbf{D}^* : ) , and δ^\dagger and \mathbf{D}^\dagger are the Gaussian derivative input image δ and its DoF

weight map, respectively (δ^\dagger : , \mathbf{D}^\dagger : ) .

In addition, θ is the targeted weight parameter in this step. The Gaussian derivative was selected because, as mentioned, focusing and defocusing within an image are identified by clear and blurred shapes, respectively, and the difference is amplified when they are differentiated. In the above equation, Step 1 is a loss term required for converting the input image into \mathbf{D}^* , and Step 2 is a loss term that determines the weight parameter when applying the Gaussian derivative filter.

After the training based on this method, we failed to obtain the expected result, and the generated images did not converge even when we increased the number of training iterations (see Fig. 9). The result of training for 15,000 epochs was an image converging to a gray image, not a DoF weight map (see Fig. 10).

The reason for this problem is that the difference between δ and \mathbf{D}^* is too large to obtain a result that converges in the intended direction after training the mapping function $f(x)$. This problem occurred in additional experiments with various images. In this study, as a solution to the problem, the input image was multiplied by the DoF weight map, and the product was used as the network training data: $\mathbf{D}^* \leftarrow \mathbf{D}^* \cdot \delta$, $\mathbf{D}^\dagger \leftarrow \mathbf{D}^\dagger \cdot \delta^\dagger$.

This study uses a residual connection-based deep neural network method to improve the algorithm. As the target is to predict the residual map, the final loss function L^* is calculated as follows: (see Eq. 8):

$$L^*(\mathbf{r}, x) = \frac{1}{2} \|\mathbf{r} - f(x)\|^2, \tag{8}$$

where \mathbf{r} denotes the residual image ($\mathbf{r} = \mathbf{D}^* - x$), and x represents the input image, which is δ . In the network training process, the loss layer is calculated using three elements: residual estimation, δ , and \mathbf{D}^* . Loss is expressed

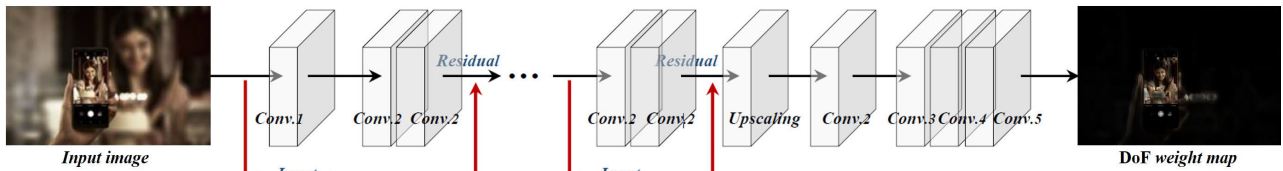


FIGURE 8. Depth-of-field extraction network architecture (red arrow: residual process).

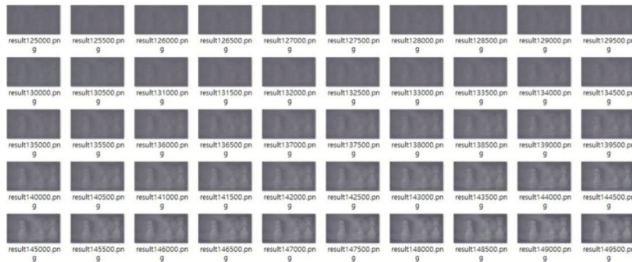


FIGURE 9. We achieved progressive results by training with the network architecture presented in Fig. 8 (input image: Fig. 10a).



FIGURE 11. Training results according to the iteration with the proposed method (left top: iteration 0, right bottom: iteration 1400). The above illustration has been arbitrarily inverted for clarity.



(a) Input image (b) Training result

FIGURE 10. The learning outcome was obtained after 15,000 iterations using the input image, employing the same network architecture as depicted in Fig. 8.

as the Euclidean distance between the map reconstructed through the network and D^* , where the reconstructed map is the sum of the input and output of the network.

This network is modeled based on the CNN, and its configuration follows (see Fig. 8). The residual compensation was performed by adding the feature map that has undergone the first CNN operation to the result of the two convolutional operations. The error caused by the convolutional operation was mitigated through residual compensation. This process is repeated 10 times; thus, 20 convolutional operations were performed. In the first cycle, only the value resulting from the first convolution is added once, and in subsequent cycles, the previous result values are added repeatedly. Next, the size is doubled through upscaling, and four convolutional operations are performed as the last step.

Fig. 11 illustrates the intermediate results of training based on the method proposed in this study. Although, as presented in Fig. 10, even 15,000 iterations produced only a gray image far from the DoF, the proposed method produced a rough DoF contour by 1400 iterations, much fewer iterations. This result indicates that the proposed method has a fast convergence

rate and can extract the DoF region. In addition, several or tens of thousands of images are generally used in the learning process, whereas the proposed method achieves an excellent learning rate with only a small dataset of 425 images due to the preprocessing method to improve the convergence rate.

III. SOLVER EXTENSIONS

The following subsections describe the solver extensions used to efficiently improve the DoF extraction algorithm discussed above through adaptive sampling. In addition, an approach is explained where only meaningful regions are extracted from the DoF region using a quadtree(Qt) and used as training data. The focused region occupies a small area compared to the entire image. These small regions are divided into patches and used as data pairs of the image-DoF weight map to be used at the network level. The resultant reduction of the data area required for learning reduces learning time and memory

A. PRELIMINARY

The Qt approach is a tree data structure in which each internal node has four children. It is an algorithm that adaptively partitions a 2D space by dividing a 2D rectangular space into four quadrants and recursively subdividing them according to a given criterion. Although data related to the leaf cell differ by application, it generally has a “minimum unit of information of interest.” In an octree, an extended version of the Qt is applied to a 3D space. Each inner node is divided into eight octants, and the cube-shaped space is recursively subdivided.

B. CLOSED-FORM FILTERING FOR ADAPTIVE SAMPLING

Before inputting data into the artificial neural network, Qt-based DoF patches are constructed using the previously



FIGURE 12. Incorrectly subdivided quadtree due to empty space (inset image: \mathbf{D}^*).

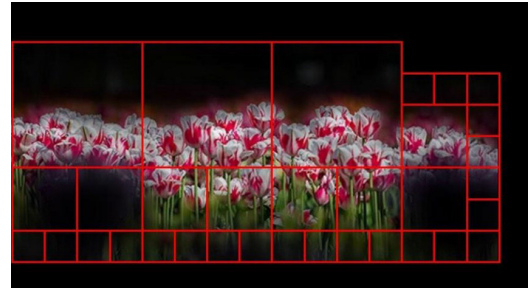


FIGURE 14. Quadtree construction with \mathbf{D}_m .

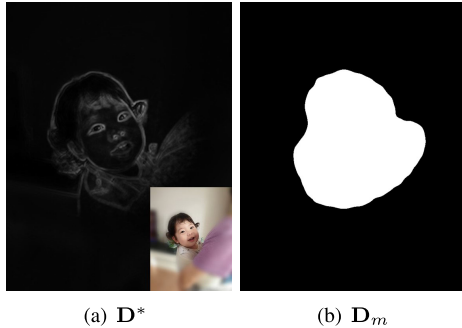


FIGURE 13. \mathbf{D}_m filtered from \mathbf{D}^* (inset image: input).

calculated \mathbf{D}^* . The only intention of performing this process is to find the DoF region. When the filter \mathbf{F} proposed in this paper is not applied, the Qt may not be properly created, as displayed in Fig. 12. Fig. 12 is the result of \mathbf{D}^* -based Qt segmentation. In monochromatic colors, the classification according to focusing and defocusing is difficult. The original image is a woman in colorful clothes and a man in a monochrome-colored T-shirt. Extraction of the DoF monochromatic region is not easy even when a focusing filter is used (see the red region in Fig. 12). This pattern is also found in \mathbf{D}^* and is more severe in the adaptive approach.

In Fig. 12, the man's top is monochromatic; therefore, almost no features, such as color scattering or spreading, are found even in defocusing. This part is not subdivided even by the Qt and remains an empty region. Thus, the Qt does not represent the DoF region correctly, which leads to the problem of a learning failure at the network level in later processes. This study proposes a novel filter \mathbf{F} to alleviate this problem (see Eq. 9):

$$\mathbf{F} = f_{bi}(f_{sp}(f_{gm}(\mathbf{D}^*))), \quad (9)$$

where f_{bi} , f_{sp} , and f_{gm} refer to binarization, sharpening, and gamma correction filters for the image, respectively.

First, f_{gm} , a gamma correction of \mathbf{D}^* is applied. This process is performed to amplify the difference between focusing and defocusing regions (see Eq. 10):

$$f_{gm}(x) = M \left(\frac{x}{M} \right)^g, \quad (10)$$

where x is the input image, which is \mathbf{D}^* in this paper. In the above equation, g represents a variable that enhances

the contrast between clear and blurred regions by applying gamma correction. In addition, M refers to the maximum value, which is set to 255 in this paper. If g is 1, the brightness changes linearly and is set to 0.85 in this study. The colors obtained through this process become clearer; however, there is a problem of blurring at the boundary or edge. To alleviate this problem, f_{sp} , an image sharpening approach using a Laplacian kernel, is applied (see Eq. 11):

$$f_{sp} = f + \alpha(f - f_{blur}), \quad (11)$$

where α can be expressed with a mask, and in this paper, a Gaussian distribution type mask is applied. Finally, through image binarization f_{bi} , \mathbf{D}_m (a mask image of \mathbf{D}^* , a DoF weight map) is extracted with a closed-type filter (see Fig. 13). To solve this problem, we applied the \mathbf{F} filter to extract a closed-type mask map, which discerns focused and defocused regions. The space partitioning method using this result and method to collect the sparse dataset are explained in the next section.

C. COLLECTING SPARSE DATASETS FROM QUADTREE-BASED DoF PATCHES

This study uses \mathbf{D}_m to calculate Qt and, as illustrated in Fig. 13a, Qt is partitioned based on the DoF region, a white part in the image. The reason to use \mathbf{D}_m to partition Qt is not to obtain pixel information but to extract meaningful information in the space. Thus, the training in the network process is performed using \mathbf{D}_p , which are patches of images representing the leaf nodes of the Qt.

The lowest node to be used in constructing a Qt is generated by the method explained above, and the tree is constructed by merging the nodes in a bottom-up manner (see Fig. 14). Before combining the generated nodes into the Qt, the nodes are classified into full density(FD) and empty density(ED) states by checking whether they have density (e.g., a DoF weight value) and, if so, comparing them with a critical value (see Fig. 15). The lowest nodes have a specified state value, and the state value of the parent node is determined by the state value of the child node (see Fig. 16a).

Each node in the tree has data, key, and state values. Data represent the density value of a node, and the key has x - and y -coordinates, indicating the node position and tree depth used when constructing the tree. Tree depth and

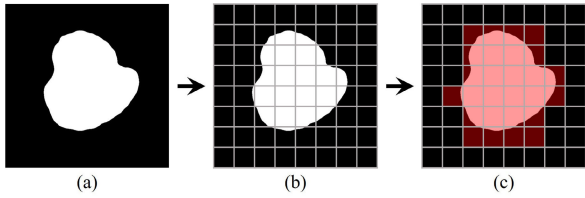


FIGURE 15. Classification as full density(FD) and empty density(ED) according to the presence or absence of density: (a) D_m , (b) patch split, (c) classification of FD (red) and ED.



(a) Visualization of the terminal node data and state



(b) Generating the state value of parent nodes in a bottom-up manner

FIGURE 16. Overview of generating the quadtree containing path-state values, where ED = empty density and FD = full density.

position are used when combining results after the network process is completed, and the state is described as FD, ED, or mixed.

In this study, the density of each patch is used as the data for the lowest node (see Fig. 16a). The depth of the lowest node is calculated using Eq. 12:

$$d = \log_2 \left(\frac{D_{width}}{N_{width}} \right) \quad (12)$$

where d refers to the depth of the current node and D_{width} and N_{width} are the widths of the entire input data and current node, respectively. A parent node is generated with four nodes in a bottom-up manner (see Fig. 16b). For the parent node, the data are the sums of those from the child nodes. The depth decreases by 1, and the position is determined by combining child nodes. Finally, the state value is determined by the child nodes: if the state values of all child nodes are the same, the value is assigned to the parent node, and if they have both FD and ED, mixed(MIX) is assigned to the state value of the parent node.

If the state values of all child nodes are the same, it is deleted. Network training is not required for those with a state value of ED. For those with FD, implementing one network training is faster than that for each child node. When this process is repeated until the root node is reached, a tree in which the state values are assigned to all nodes is generated. After the tree is completed, the data and key values for all FD nodes are collected (see Fig. 17), and this dataset is used for network training.

Fig. 18a is a dataset collected from a Qt, and network training is performed using this information. As described above, this value is just for obtaining spatial information,

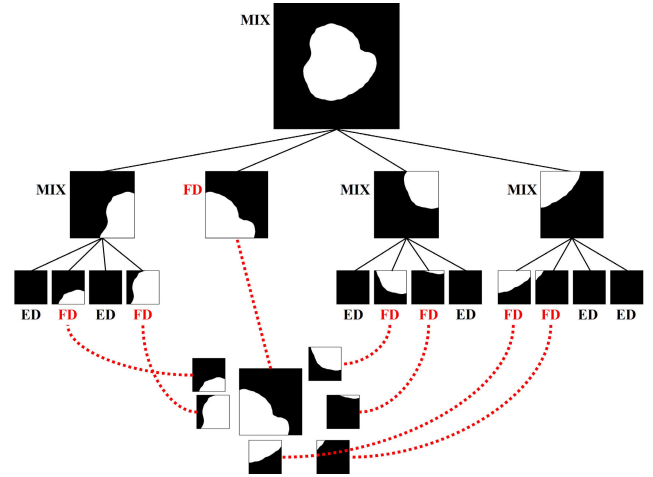
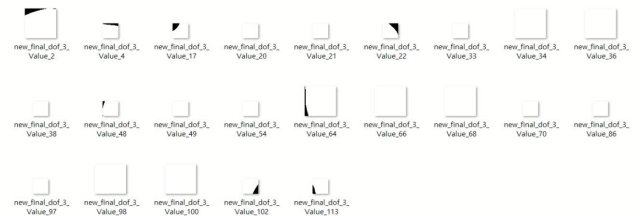
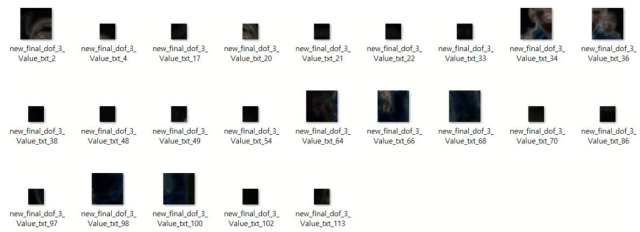


FIGURE 17. Example of quadtree nodes for collecting depth-of-field(DoF) patches (dotted red line: final dataset, D_m patches).



(a) D_m patches



(b) D^* patches

FIGURE 18. Collected depth-of-field(DoF) mask patches (D_m) and weight maps (D^*) in the leaf nodes of the quadtree.

and D^* patches, which are image data corresponding to the location of these patches, are used for training (see Fig. 18b). The number of datasets used in this approach is less than that of the original data; thus, the training process is efficiently optimized. Constructing Qt required 1 second per image, and when using the 425 data points for training, it took 18 hours to construct Qt. However, without using Qt, it took 99 hours for the training process.

IV. IMPLEMENTATION

The test was implemented in the following environment: Intel i7-7700k 4.20 GHz CPU, 32 GB of RAM, with an NVIDIA GeForce GTX 1080 Ti GPU.

A. DETAILS OF DoF EXTRACTION NETWORKS

In this study, a DoF extraction network is designed based on a neural network. In this paper, we designed a network structure to implement the hyperparameters used for training,

TABLE 1. Configuration of depth-of-field(DoF) extraction convolutional neural networks(CNNs).

	CNN r1	CNN r2	CNN r3	CNN r4	CNN r5	CNN r6	CNN r7
Transpose (...)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	(2,2)	-
#x(w, h, d)	(8,8,64)	(16,16,64)	(32,32,64)	(64,64,64)	(128,128,64)	(256,256,64)	-
concat (...)	(input x, CNN 5)	(input x, CNN 4)	(input x, CNN 3)	(input x, CNN 2)	(input x, CNN 1)	-	-
#x(w, h, d)	(8,8,64+512)	-	-	-	-	-	-
[weight],[bias]	$5 \times 5 \times 64, 64$	$5 \times 5 \times 64, 64$	$5 \times 5 \times 64, 64$	$5 \times 5 \times 64, 64$	$5 \times 5 \times 64, 64$	$5 \times 5 \times 64, 64$	$5 \times 5 \times 3, 3$
Num. CNN	CNN r1-4	CNN r2-4	CNN r3-4	CNN r4-2	CNN r5-2	CNN r6-2	CNN r7-1
#x(w, h, d)	(8,8,64)	(16,16,64)	(32,32,64)	(64,64,64)	(128,128,64)	(256,256,64)	(256,256,3)

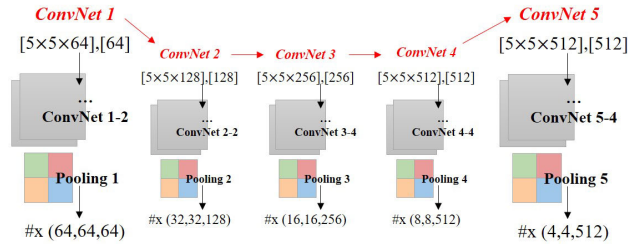


FIGURE 19. Feature extraction (input: $x(128, 128, 3)$, output: $x(4, 4, 512)$, [weight], [bias], #x (width, height, depth)).

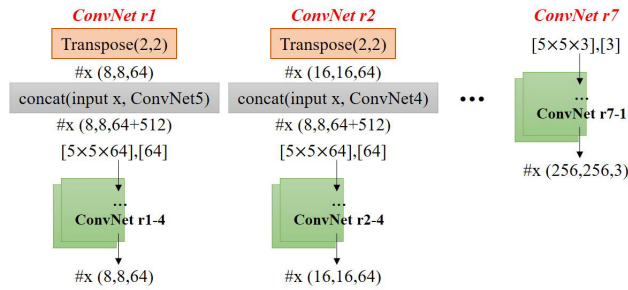


FIGURE 20. Reconstruction (input: $x(4, 4, 512)$, output: $x(256, 256, 3)$).

which were batch size of 128, learning rate of $1e-4$, and stride of 1. The DoF extraction network consists of two main steps: feature extraction and reconstruction. To distinguish the sharp and blurry areas represented in the DoF, a CNN-based approach is used. Specifically, the CNN is trained to find the weight map of the sharply represented area in the DoF from the input image. This process involves extracting patches from the input image and representing each patch as a high-dimensional vector, which constitutes a set of feature maps that are used during training. During the training process, the filters in the CNN convolve through the image to extract features and learn the weight map of the sharply represented area in the DoF. For example, when the input image is a three-channel image with 128×128 resolution, the input x is $x(128, 128, 3)$.

In addition, CNN 1 consists of two layers and uses the ReLU activation function. The output value of CNN 1 is the input of CNN 2. Fig. 19 depicts the overall feature extraction process. In this figure, CNN 1-2 indicates that two CNN layers are used in the CNN 1 process, and CNN 5-4 indicates that four layers are used in the CNN 5 process.

In the reconstruction stage, the image is reconstructed by applying transposed convolution to the feature extraction

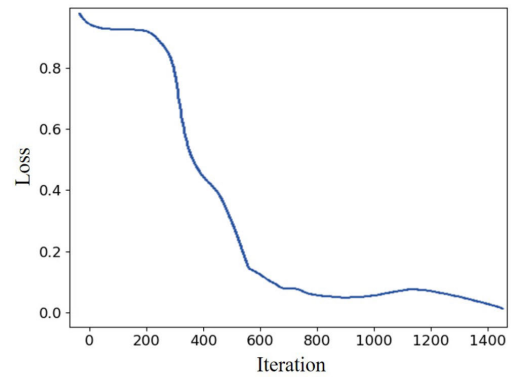


FIGURE 21. Training loss.

result (see Fig. 20). Table 1 details the CNN r1-CNN r7. This network was implemented in TensorFlow. Adam was used as the optimizer, and optimization of 1400 epochs was performed (see Fig. 21).

It is common to use a 3×3 size filter, but we used a 5×5 size filter for more effective receptive field coverage in this paper. Through experimentation, we confirmed that this approach yields slightly better performance for the current task.

B. DATASET DETAILS

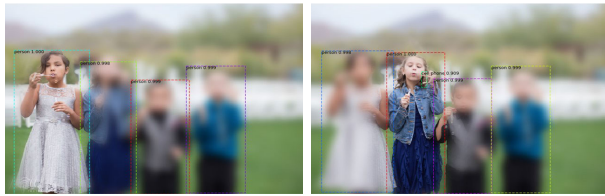
In this study, to collect data for training the DoF, 425 images containing DoF data were obtained through web crawling and used for network training. In the testing stage, in addition to images obtained through web crawling, those images whose DoF effect was artificially edited using such tools as the Photo Editor application were also used.

V. RESULTS

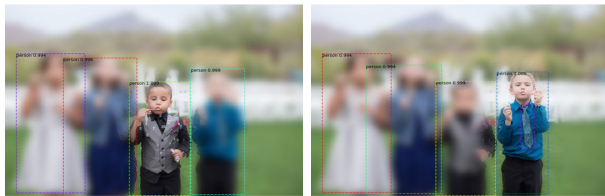
The comparisons were made in six scenarios to analyze the proposed method under various conditions: 1) DoF region detection, 2) object recognition, 3) NPR, 4) viewport tracking, 5) OCR, and 6) adaptive sampling.

A. DEPTH-OF-FIELD REGION DETECTION

An experiment was conducted to extract the DoF region from the input image using the proposed method, and Fig. 40 presents the result: D^* obtained using only the proposed network architecture without adaptive sampling. The final result is (D_r^*) generated by converting to rainbow colors for better visualization of the DoF weight. As depicted in the



(a) Accuracy: 1.0, 0.998, 0.999, (b) Accuracy: 0.998, 1.0, 0.999, and 0.999, recognition: person, per-son, person, person



(c) Accuracy: 0.994, 0.996, 1.0, (d) Accuracy: 0.994, 0.994, 0.999, and 0.999, recognition: person, per-son, person, person

FIGURE 22. Scene 1 (object recognition without the proposed method: object accuracy left to right).

figure, the DoF region is reliably extracted in most results. Such excellent performance is hardly found in previous studies that edit or estimate depth. The goal of the existing approaches was to extract the actual depth, whereas the study result is unaffected by the actual depth because the result depends on the camera focusing position determined by the user. Nevertheless, stable results were obtained.

B. OBJECT RECOGNITION

This subsection reviews the result of object recognition using DoF weights. As mentioned in Section **Problem Statement**, when a person views an object, the focused object is clear, and the defocused object is blurred. These characteristics are also found in camera focusing.

Fig. 22 presents the results of the object recognition performed using YOLO [40]. Although the camera focus moves from left to right, the level of person recognition, as indicated in the caption, exhibits almost no difference. Focusing and defocusing effects that appear when a person or a camera views the image are not applied to this result. On the contrary, the proposed method demonstrates that only the focused object among several people is extracted accurately (see Fig. 23). This result is maintained as the viewpoint changed from left to right, and object recognition accuracy was close to 1.0.

Fig. 24 displays the experimental results on recognizing animals rather than people. Similar to the results for the people, the DoF expressed in the image was not considered in object recognition. The accuracy values in the captions also exhibit a high value for defocused objects. This result is because the ROI of the person is not effectively considered, whereas the proposed method indicates a result in which only the focused objects are recognized (see Fig. 25). In Scene 2,



(a) Accuracy: 1.0, recognition: per-son, x, x, x (b) Accuracy: 0.998, recognition: x, person, x, x



(c) Accuracy: 0.999, recognition: x, x, person, x (d) Accuracy: 0.999, recognition: x, x, person

FIGURE 23. Scene 1 (object recognition with the proposed method).



(a) Bird (accuracy): 0.995, 0.999, (b) Bird (accuracy): 0.96, 0.906, 1.0, 0.994, 0.9, and 0.987 0.996, 0.917, and 0.987



(c) Bird (accuracy): omit (d) Bird (accuracy): omit

FIGURE 24. Scene 2 (object recognition without the proposed method).

unlike Scene 1, object recognition is performed accurately even when focusing is located amid objects (see Fig. 25c).

C. VIEWPORT TRACKING

One of the many areas where DoF can be used is user viewport tracking. That is, the following various analyses may be performed. In what order did the subject view the different content? What about changing the subject's viewpoint? What object is the subject primarily viewing? How long does the subject's gaze stay on each object?

Fig. 26 presents the results of DoF-based viewport tracking using the proposed method. As this is not the result of object recognition-based tracking but the result of DoF-based tracking that considers the user's focal point, an ROI analysis is allowed according to the user's change in viewpoint and the object of interest to the user. These results cannot be obtained

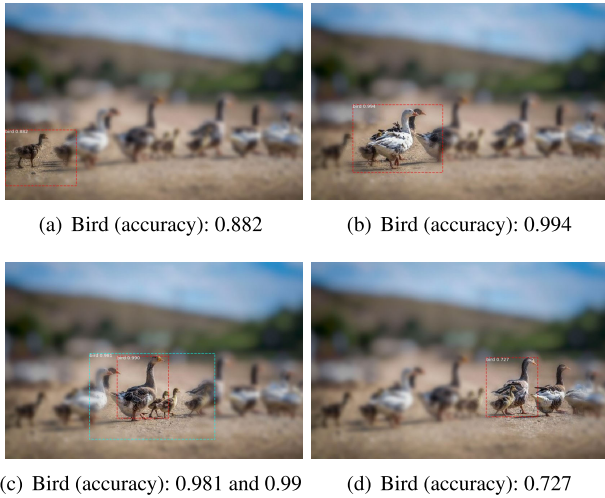


FIGURE 25. Scene 2 (object recognition with the proposed method).



FIGURE 26. Viewport tracking with the proposed method (red line: change in viewpoint).

with object detection and recognition algorithms alone. This technique may be used for various application analyses, such as the additional scenarios below.

1) ANALYSIS BY OBJECT RECOGNITION ORDER

Fig. 26a reveals the result of object recognition according to the user’s viewpoint. As the viewpoint moves from left to right, the following relationship results: p1→p2→p3→p4. Through this analysis, the order in which the subject views objects or content can be identified.

2) CUMULATIVE TIME ANALYSIS FOR EXAMINING THE OBJECT OF INTEREST

The proposed method can determine which object the subject viewed for a longer period through a cumulative time

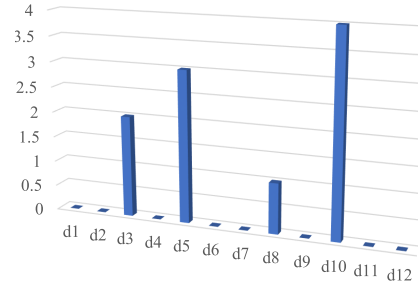


FIGURE 27. Time log in Fig. 26b.

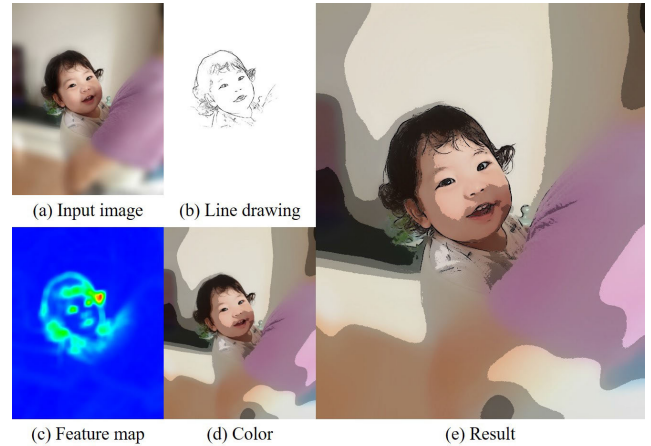


FIGURE 28. Scene 3 (as a result of applying the proposed method to the NPR technique [30]).

analysis Fig. 26b presents the results generated through the accumulation of the object recognition time, measured from when the object is recognized to when it is released. This analysis is impossible when only object recognition is applied, and the DoF analysis is necessary to identify which object has the subject’s focus. of viewpoints for each object. Fig. 27 illustrates the log chart of the object recognition time. Although we displayed just the ROI of the viewpoint in time in the image, this technique is expected to be extended to the field of forensic investigation, where the user’s tension or concentration is examined scientifically by analyzing the shaking or the direction of the viewpoint.

D. NON-PHOTOREALISTIC RENDERING

In addition, NPR is a rendering technique widely used in 3D [28], [29] and 2D images [26], [27]. In the animation field, cartoon-style expression using the motion exaggeration technique is attempted, and in the video field, cartoon effects are expressed through color simplification and outline sketches. As NPR filtering of the entire image is applied in most methods, the characteristics of a specific ROI are not considered. Recently, Kim et al. presented a DoF-based NPR technique to solve this problem [30], and in this study, the NPR result is generated by integrating this technique with the calculated DoF weight map.

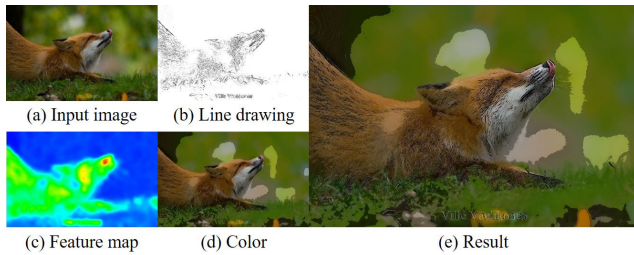


FIGURE 29. Scene 4 (as a result of applying the proposed method to the NPR technique [30]).

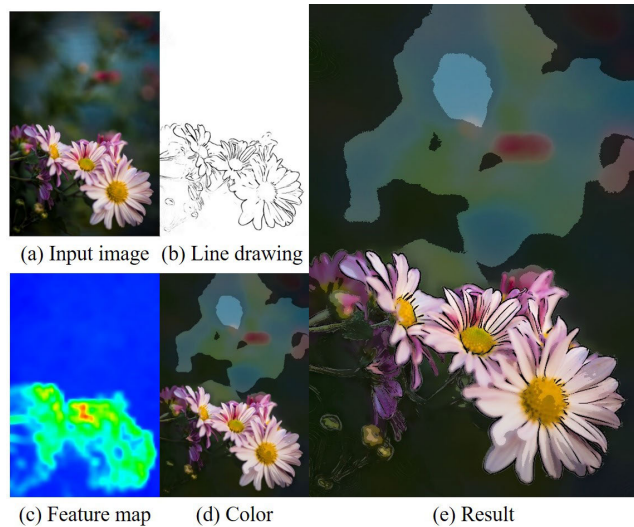


FIGURE 30. Scene 5 (as a result of applying the proposed method to the NPR technique [30]).

Fig. 28 depicts the NPR result generated using the DoF weight map as the NPR feature map. The NPR features around the focused baby face reveal that the color and magnitude of the line drawing are controlled. Fig. 29 is the result of experiments performed in a more complex scene. The focused object of the input image is a fox, and the DoF weight map excellently captures the characteristics of the focused object (see Fig. 29c). The accuracy of the DoF weight map is also clearly confirmed in the line drawing and color (see Fig. 29b and 29c). When the proposed technique, which infers DoF based on an artificial neural network, is applied to the NPR technique, the results of all analyses are also stable. This excellent performance was maintained even in the flower image (see Fig. 30).

E. OPTICAL CHARACTER RECOGNITION

Moreover, OCR is a technique that recognizes characters from scanned documents and actual images and videos and serves in various fields [31], [32]. Text recognition is as important as object recognition, and it is easy to expand into research fields, such as natural language processing, because the meaning of a sentence is identified through it. However, just the recognition of characters, such as English [33],

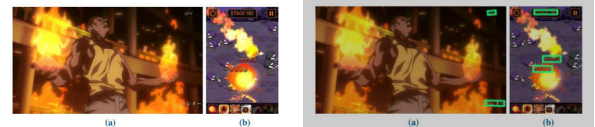


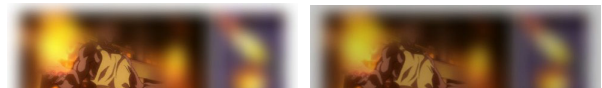
FIGURE 31. Various fire effects in animation and game (a: animation ZETMAN; b: mobile game Raising Fire Magician).

The fire effect based on a still frame can be automatically generated using a procedural approach [11], [13]. However, the procedural approach makes it difficult to add or edit effects other than predefined fire effects. A simulation-based approach can produce realistic and very detailed flame effects. The shape and flow of the flame can be intuitively controlled by placing the fuel to follow the target shape, burning it, and then inducing the flame in the desired direction. However, because of the large amount of computation, simulating fire simulation in low-resolution is required to

The fire effect based on a still frame can be automatically generated using a procedural approach [11], [13]. However, the procedural approach makes it difficult to add or edit effects other than predefined fire effects. A simulation-based approach can produce realistic and very detailed flame effects. The shape and flow of the flame can be intuitively controlled by placing the fuel to follow the target shape, burning it, and then inducing the flame in the desired direction. However, because of the large amount of computation, simulating fire simulation in low-resolution is required to

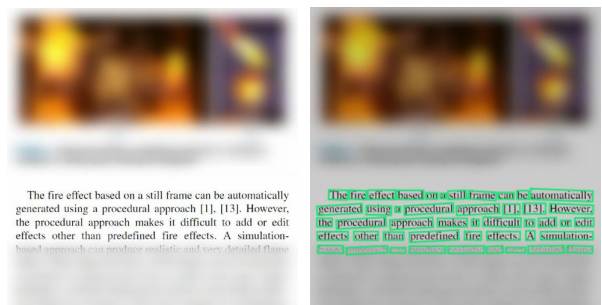
(a) Input image (b) Result

FIGURE 31. Optical character recognition [31] results calculated on images without the depth of field.



(a) Input image (b) Result

FIGURE 32. Optical character recognition [31] results calculated on images with the depth of field.



(a) Input image (b) Result

FIGURE 33. Optical character recognition [31] results calculated on images with the anisotropic depth of field.

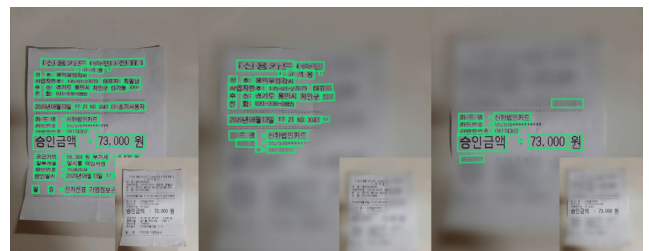


FIGURE 34. Various optical character recognition [31] results with the proposed method.

Chinese [34], [35], Japanese [36], and Korean [37], [38] from a clean input image was sought, and the approach considering the DoF was not attempted. This study performed

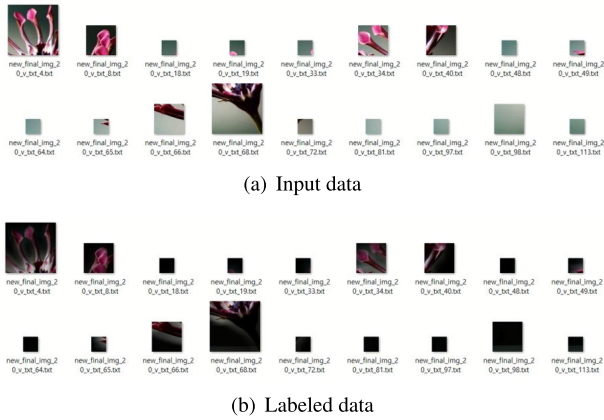


FIGURE 35. Refined dataset using the quadtree.

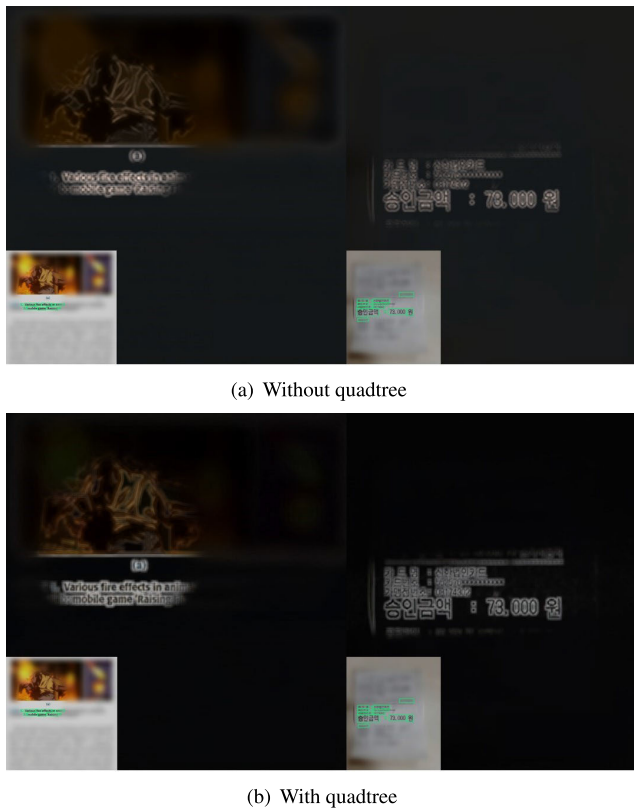


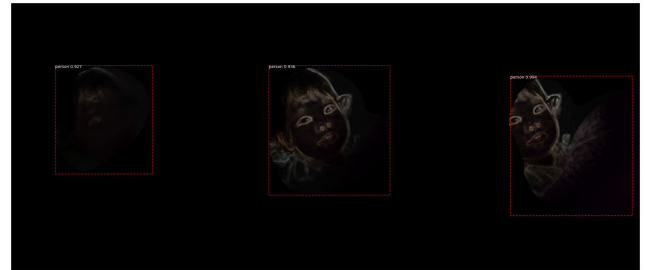
FIGURE 36. Quadtree-based D^* tested in Figs. 33 and 34 (inset image: input data).

the test by integrating the calculated DoF weight map with the conventional OCR algorithm technique.

Fig. 31 is the result of performing the OCR algorithm on the captured image from a portable document format(PDF) file, indicating that the text included in the image is stably extracted. However, most methods consider text extraction from clear images and do not consider the DoF, which is the user's actual ROI. As previously demonstrated, the DoF affects object detection and recognition and the text on which the user focuses. Fig. 32 is the result of the combination of the

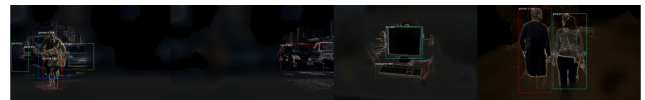


(a) With quadtree (accuracy: 0.823, 0.931, and 0.997)

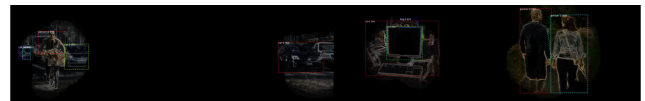


(b) Without quadtree (accuracy: 0.927, 0.936, and 0.994)

FIGURE 37. Object recognition using quadtree-based D^* .



(a) With quadtree



(b) Without quadtree

FIGURE 38. Results of object recognition using quadtree-based D^* .

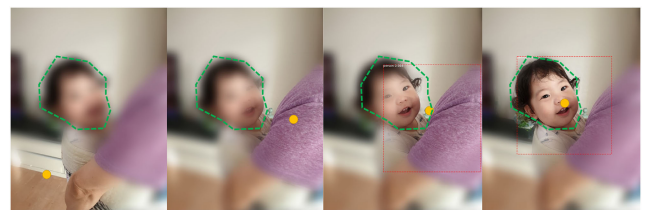


FIGURE 39. Results according to the relationship between the object to be recognized and the focusing position (dotted green line: recognized object, orange circle: focusing position).

proposed method and OCR algorithm, and unlike Fig. 31, the text area extracted by focusing is presented. Fig. 33 is the result of applying the anisotropic artificial DoF and reveals the OCR result considering the DoF despite applying a rectangle-shaped blurring filter. Fig. 34 is the result of another experiment using the proposed method and reveals that stable results are produced in isotropic artificial and anisotropic filtering. Although this study did not present a new OCR algorithm, it indicates that the proposed method

can be stably integrated into the existing OCR technique and that extracting results considering the DoF is easy.

F. ADAPTIVE SAMPLING

This subsection describes efficiency improvement through the adaptive extension of the DoF network based on Qt. Fig. 35 illustrates the input and labeled data defined according to the node position of Qt. In the experiment using the image in Fig. 11, training took 18 hours when using Qt and four days and three hours when not using Qt.

Fig. 36 compares the results with and without Qt. These results indicate that applying Qt leads to a better outcome in terms of memory or speed.

Fig. 37, the result of object recognition, illustrates that object recognition is stably performed even when Qt was applied, and the accuracy was almost similar. Although Qt improves efficiency, problems have been observed regarding accuracy. In Fig. 37a and 37b, the object accuracy values of the left two images are relatively high at 0.823 and 0.927, respectively. However, the object was recognized as a “cat” in Fig. 37a and as a “person” in Fig. 37b. This difference is because almost no focusing occurs in the input data, and most of the area around the recognition target object is defocused. Nevertheless, the person is recognized correctly in Fig. 37a. Although the focused region was cropped and used by minimizing the input data size using Qt, the defocused region decreased in the process, leading to a difference in accuracy. However, this problem occurs when defocusing is strongly expressed around the recognition target object, and the object recognition results in most other situations are stable. Not all scenes showed a decrease in accuracy when using Qt. This feature was observed in only a few scenes. The reason for this is that Qt divides the image based on the focusing area, which may result in the omission of relatively defocused areas, leading to a decrease in accuracy. Fig. 38 has the test results in various scenes, confirming stable results in most situations.

DISCUSSION

Comparison of the Relationship Between the DoF and Object Recognition: A validation test was performed by applying the proposed technique to various applications. The recognition rate in object recognition varies depending on the location and size of the defocused region, and this section discusses the relationship between the DoF and object recognition.

Fig. 39 details the test results for recognition accuracy according to the relationship between the recognition target and focusing position. The recognition target is not recognized when it is far from the focusing position, and more accurate recognition is performed when the distance is shorter. The two images on the right of Fig. 39 are recognized as a “person,” and their accuracy is 0.984 and 1.0, respectively. In this paper, we chose CNN over GAN for the following reasons: While GANs may be effective for DoF generation, the ambiguity between smooth texture and out-of-focus blurs, as well as the complexity of the model, can

become obstacles. We aimed to propose a simple yet effective method using filter-based low-cost techniques.

VI. CONCLUSION AND FUTURE WORK

This study presents a CNN-based method to extract DoF regions from images efficiently and improves efficiency through solver extensions based on adaptive grids. The metadata required for training was calculated through a cross-correlation filter, and the DoF region could be extracted rapidly with fewer data and iterations by presenting a method to improve the convergence rate in the training process. This study demonstrated that the proposed method, which considers the user’s focusing/defocusing features, such as the ROI of an image, can be applied to various applications, such as DoF region detection, object recognition, NPR, viewport tracking, and adaptive sampling for a stable analysis. The reason we used Gaussian filtering in this paper is twofold: 1) to efficiently distinguish the focusing and defocusing regions, and 2) to improve learning efficiency without diverging at the network stage. In this process, if we were to use a bilateral filter, it would play a role in reducing noise, and we believe it could be an effective alternative to the anisotropic kernel used in this paper.

In this paper, the proposed method is evaluated based on whether the blurred and sharp areas in the DoF are clearly distinguished visually, since it is difficult to apply PSNR (Peak Signal-to-Noise Ratio), which is commonly used to measure accuracy in image super-resolution. Although there are methods such as PSN-HSV, PSNR-HVS-M, and SSIM (Structural Similarity Index Map) that can measure accuracy based on MSE, it is difficult to apply them intuitively, and it is challenging to restore losses in high texture details. Although there are approaches to estimate and control the depth map, the method proposed in this paper, which extracts the DoF area, has not been seen before. While it may be possible to compare the proposed method with modified depth estimation approaches, it would require algorithmic modifications and is therefore not included in this research. We will design a precise comparison metric and prepare for future comparisons with various methods.

Nevertheless, this study has several limitations. Although performance improved by reducing the training data size in the optimization process based on Qt, the object recognition accuracy slightly decreased because only the focusing region is considered in the tree construction. We adopted this approach because the defocused region is approximated almost as a black region due to the nature of the DoF weight map. Although we did not directly experiment with Canonical Correlation Analysis (CCA), a probabilistic analysis method that determines the linear relationship between two datasets, we believe that algorithm extension using this method could also enable the discovery of DoF. This problem is expected and will be solved using an improved padding technique in future work.

REFERENCES

- [1] P. Gargallo and P. Sturm, "Bayesian 3D modeling from images using multiple depth maps," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR05)*, pp. 885–891.
- [2] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.
- [3] C. Häne, L. Heng, G. H. Lee, F. Fraundorfer, P. Furgale, T. Sattler, and M. Pollefeys, "3D visual perception for self-driving cars using a multi-camera system: Calibration, mapping, localization, and obstacle detection," *Image Vis. Comput.*, vol. 68, pp. 14–27, Dec. 2017.
- [4] K. Müller, P. Merkle, and T. Wiegand, "3-D video representation using depth maps," *Proc. IEEE*, vol. 99, no. 4, pp. 643–656, Apr. 2011.
- [5] C. Häne, C. Zach, J. Lim, A. Ranganathan, and M. Pollefeys, "Stereo depth map fusion for robot navigation," in *Proc. IEEE/RSI Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 1618–1625.
- [6] G. Rafiee, S. S. Dlay, and W. L. Woo, "Region-of-interest extraction in low depth of field images using ensemble clustering and difference of Gaussian approaches," *Pattern Recognit.*, vol. 46, no. 10, pp. 2685–2699, Oct. 2013.
- [7] J. Park and C. Kim, "Extracting focused object from low depth-of-field image sequences," *Vis. Commun. Image Process.*, vol. 6077, Jan. 2006, Art. no. 607710.
- [8] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. R. S. Müller, and Y. LeCun, "Learning long-range vision for autonomous off-road driving," *J. Field Robot.*, vol. 26, no. 2, pp. 120–144, 2009.
- [9] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2366–2374.
- [10] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, "Deeper depth prediction with fully convolutional residual networks," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 239–248.
- [11] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5162–5170.
- [12] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
- [13] Z. Li and N. Snavely, "MegaDepth: Learning single-view depth prediction from internet photos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2041–2050.
- [14] P. P. Srinivasan, R. Garg, N. Wadhwa, R. Ng, and J. T. Barron, "Aperture supervision for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6393–6401.
- [15] S. Suwajanakorn, C. Hernandez, and S. M. Seitz, "Depth from focus with your mobile phone," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3497–3506.
- [16] J. T. Barron, A. Adams, Y. Shih, and C. Hernández, "Fast bilateral-space stereo for synthetic defocus," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4466–4474.
- [17] H. Ha, S. Im, J. Park, H.-G. Jeon, and I. S. Kweon, "High-quality depth from uncalibrated small motion clip," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5413–5421.
- [18] N. Joshi and C. L. Zitnick, "Micro-baseline stereo," Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2014-73, 2014, p. 8.
- [19] F. Klose, O. Wang, J.-C. Bazin, M. Magnor, and A. Sorkine-Hornung, "Sampling based scene-space video processing," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 1–11, Jul. 2015.
- [20] P. Haeblerli and K. Akeley, "The accumulation buffer: Hardware support for high-quality rendering," *ACM SIGGRAPH*, vol. 24, no. 4, pp. 309–318, 1990.
- [21] S. Lee, E. Eisemann, and H.-P. Seidel, "Real-time lens blur effects and focus control," *ACM Trans. Graph.*, vol. 29, no. 4, p. 1, Jul. 2010.
- [22] M. Kraus and M. Strengert, "Depth-of-field rendering by pyramidal image processing," *Comput. Graph. Forum*, vol. 26, no. 3, pp. 645–654, Sep. 2007.
- [23] S. Lee, G. J. Kim, and S. Choi, "Real-time depth-of-field rendering using anisotropically filtered mipmap interpolation," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 3, pp. 453–464, May 2009.
- [24] Y. Yang, H. Lin, Z. Yu, S. Paris, and J. Yu, "Virtual DSLR: High quality dynamic depth-of-field synthesis on mobile platforms," *Electron. Imag.*, vol. 28, no. 18, pp. 1–9, Feb. 2016.
- [25] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.
- [26] B. Gooch, G. Coombe, and P. Shirley, "Artistic vision: Painterly rendering using computer vision techniques," in *Proc. 2nd Int. Symp. Non-photorealistic animation rendering*, Jun. 2002, p. 83.
- [27] J. Hays and I. Essa, "Image and video based painterly animation," in *Proc. 3rd Int. Symp. Non-Photorealistic Animation Rendering*, Jun. 2004, pp. 113–120.
- [28] J. Fischer, D. Bartz, and W. Straber, "Stylized augmented reality for improved immersion," in *Proc. IEEE Proc. VR Virtual Reality*, 2005, pp. 195–202.
- [29] R. D. Kalnins, L. Markosian, B. J. Meier, M. A. Kowalski, J. C. Lee, P. L. Davidson, M. Webb, J. F. Hughes, and A. Finkelstein, "WYSIWYG NPR: Drawing strokes directly on 3D models," in *Proc. 29th Annu. Conf. Comput. Graph. Interact. Techn.*, Jul. 2002, pp. 755–762.
- [30] J.-H. Kim and J. Lee, "Layered non-photorealistic rendering with anisotropic depth-of-field filtering," *Multimedia Tools Appl.*, vol. 79, nos. 1–2, pp. 1291–1309, Jan. 2020.
- [31] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character region awareness for text detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9365–9374.
- [32] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? Dataset and model analysis," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 4715–4723.
- [33] R. Parthiban, R. Ezhilarasi, and D. Saravanan, "Optical character recognition for English handwritten text using recurrent neural network," in *Proc. Int. Conf. Syst., Comput., Autom. Netw. (ICSCAN)*, Jul. 2020, pp. 1–5.
- [34] H. Yu, J. Chen, B. Li, J. Ma, M. Guan, X. Xu, X. Wang, S. Qu, and X. Xue, "Benchmarking Chinese text recognition: Datasets, baselines, and an empirical study," 2021, *arXiv:2112.15093*.
- [35] B. Liu, X. Xu, and Y. Zhang, "Offline handwritten Chinese text recognition with convolutional neural networks," 2020, *arXiv:2006.15619*.
- [36] B. Zhu, X.-D. Zhou, C.-L. Liu, and M. Nakagawa, "A robust model for on-line handwritten Japanese text recognition," *Int. J. Document Anal. Recognit. (IJ DAR)*, vol. 13, no. 2, pp. 121–131, Jun. 2010.
- [37] S. Ilyuhin, A. Sheshkus, and V. L. Arlazarov, "Recognition of images of Korean characters using embedded networks," in *Proc. 12th Int. Conf. Mach. Vis. (ICMV)*, Jan. 2020, Art. no. 1143311.
- [38] H. Eun, J. Kim, J. Kim, and C. Kim, "Fast Korean text detection and recognition in traffic guide signs," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2018, pp. 1–4.
- [39] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1800–1807.
- [40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.
- [41] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1141–1151, Oct. 2002.
- [42] M. M. Tiwari, I. Misra, S. M. Moorthi, and D. Dhar, "An improved IHS image fusion algorithm using medoid intensity match and bilateral filter," in *Proc. IEEE Int. India Geosci. Remote Sens. Symp. (InGARSS)*, Dec. 2021, pp. 500–503.
- [43] H. Hotelling, "Relations between two sets of variates," in *Breakthroughs in Statistics: Methodology and Distribution*. New York, NY, USA: Springer, 1992, pp. 162–190.
- [44] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Image deblurring with blurred/noisy image pairs," *ACM Trans. Graph.*, vol. 26, no. 99, p. 1, Jul. 2007.
- [45] J. Back, B.-S. Hua, T. Hachisuka, and B. Moon, "Self-supervised post-correction for Monte Carlo denoising," in *Proc. ACM SIGGRAPH Conf.*, Aug. 2022, pp. 1–8.
- [46] I. Misra, M. K. Rohil, M. M. Subbiah, and D. Dhar, "EPOCH: Enhanced procedure for operational change detection using historical invariant features and PCA guided multivariate statistical technique," *Geocarto Int.*, vol. 37, no. 25, pp. 9369–9391, Dec. 2022.
- [47] B. J. Meier, "Painterly rendering for animation," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, Aug. 1996, pp. 477–484.
- [48] J. P. Collomosse and P. M. Hall, "Painterly rendering using image saliency," in *Proc. 20th Eurographics U.K. Conf.*, pp. 122–128.

•••