Adversarial Patch Attacks on Monocular Depth Estimation Networks

KOICHIRO YAMANAKA,1 RYUTAROH MATSUMOTO,2,3 (Member, IEEE), KEITA TAKAHASHI,1 (Member, IEEE), AND TOSHIAKI FUJII,1 (Member, IEEE)

1Graduate School of Engineering, Nagoya University, Nagoya 464-8603, Japan
2Department of Information and Communications Engineering, Tokyo Institute of Technology, Tokyo 152-8550, Japan
3Department of Mathematical Sciences, Aalborg University, 9220 Aalborg, Denmark

Corresponding author: Koichiro Yamanaka (koichiro.ymnk@fujii.nuee.nagoya-u.ac.jp)

ABSTRACT Thanks to the excellent learning capability of deep convolutional neural networks (CNN), monocular depth estimation using CNNs has achieved great success in recent years. However, depth estimation from a monocular image alone is essentially an ill-posed problem, and thus, it seems that this approach would have inherent vulnerabilities. To reveal this limitation, we propose a method of adversarial patch attack on monocular depth estimation. More specifically, we generate artificial patterns (adversarial patches) that can fool the target methods into estimating an incorrect depth for the regions where the patterns are placed. Our method can be implemented in the real world by physically placing the printed patterns in real scenes. We also analyze the behavior of monocular depth estimation under attacks by visualizing the activation levels of the intermediate layers and the regions potentially affected by the adversarial attack.

INDEX TERMS Adversarial attack, monocular depth estimation, CNN.

I. INTRODUCTION

Estimating pixel-wise depth from 2-D images has become increasingly important with the recent development of autonomous driving, augmented realities (AR), and robotics. A large body of previous work has been devoted to depth estimation from stereo or more than two images [1]–[4]. At the same time, monocular depth estimation [5]–[8], in which depth is estimated from a single image, has attracted attention due to its less demanding hardware requirements. Monocular depth estimation has been greatly enhanced by the excellent learning capability of deep convolutional neural networks (CNN). As a result, current state-of-the-art results with monocular depth estimation are quite impressive, and seemingly comparable to those with stereo methods (see Fig. 1, where (a) is the input image and (b) is the depth estimated by Guo et al. [7]). However, monocular depth estimation is essentially an ill-posed problem because a monocular image alone does not contain sufficient physical cues for scene depth. Instead of using the physical cues, these methods seem to rely on implicit knowledge (e.g., the color, vertical position, or shadows) that are learned from the training dataset [12]. We argue that monocular depth estimation depends too much on non-depth features in the given image, which makes it quite vulnerable to attacks.

To reveal the limitation mentioned above, we propose a method of adversarial patch attack for CNN-based monocular depth estimation. Specifically, we generate artificial patterns (adversarial patches) that can fool the target methods into estimating an incorrect depth for the regions where the patterns are placed. Figure 1(c) shows an example of our adversarial patches superimposed on the input image. As shown in (d), Guo et al.’s method [7] failed to estimate correct depth in the region where the patch was located; closer depth values were obtained than the original result in (b), as was intended with our design for this pattern. In this case, the attack was conducted in a digital manner; we digitally manipulated the pixel values of the input image to superimpose the patch. Our method can also be implemented in the real world, and we have achieved similar effects by physically placing the printed patterns in a real scene.

Moreover, to further analyze the behavior of monocular depth estimation under attacks, we visualize the
A number of studies have also been conducted on “physical adversarial attacks” that can be implemented in the real world, e.g., by placing printed patterns in a target scene. These patterns are not necessarily designed to be imperceptible to the human eye depending on the applications [24]–[27]. Kurakin et al. [28] demonstrated that images with adversarial patterns for a classification task created in [15] would remain adversarial when they are printed and captured by cameras. Athalye et al. [29] extended this idea to 3D physical adversarial objects. Eykholt et al. [30] showed that stop signs can be misclassified if various stickers are placed on top of them. Their adversarial objects were designed to be indistinguishable to the human eye, similarly to the case with “digital adversarial attacks”.

In a similar vein, Brown et al. [24] took a small designed patch (adversarial patch), which was clearly visible to the eye, and placed it in target scenes to induce errors in a classification task. Their patches were designed to be placed anywhere in an input image. Moreover, as their patches are independent of target scenes, they can be used for “physical adversarial attacks” without prior knowledge of lighting conditions, camera angles, or other objects in the target scene. This is not trivial, as a pattern located in the real world usually receives a series of transformations (e.g., geometric transform, digitization, and color gamut transform) before it is recorded in a digital image, which would invalidate the effect as an adversarial pattern [31]. Following [24], adversarial patches have been used in several tasks such as face recognition [25], object detection [26] and optical flow estimation [27]. Komkov and Petiushko [25] attacked face recognition by sticking an adversarial patch on a hat. Ranjan et al. [27] conducted an adversarial patch attack on optical flow that is, to our knowledge, the first work to apply adversarial patch attacks to a regression problem. Note that our method can also be located in the context of adversarial patch attack for a regression problem.

2) PHYSICAL ADVERSARIAL ATTACKS

B. MONOCULAR DEPTH ESTIMATION

Monocular depth estimation refers to the process of predicting pixel-wise depth from a single image. As a seminal work, Eigen et al. [32] proposed a multi-scale CNN architecture that can produce pixel-wise depth estimation from a single image. Unlike other previous works in single image depth estimation [33]–[38], their network did not rely on hand crafted features. Since then, significant improvements have been made by using techniques such as incorporation of strong scene priors for surface normal estimation [36], conditional random fields [39], conversion from a regression problem to a classification problem [40], and a quantized ordinal regression problem [6]. Lee et al. [8] achieved state-of-the-art result by introducing novel local planar guidance layers located at multiple stages in the decoding phase. All of the methods mentioned above are trained in a supervised manner, where ground-truth depth taken by RGB-D cameras or 3D laser scanners are required to train the networks.
Recently, semi-supervised and unsupervised methods have also been proposed. As examples of the semi-supervised approach, Chen et al. [41] used relative depth information and Kuznietsov et al. [42] used sparse depth data obtained from LiDAR. As for the unsupervised approaches, they require only rectified stereo image pairs to train networks. Xie et al. [43] proposed Deep3D to create a new right view from an input left image using depth image-based rendering [44], where a disparity map was estimated as an intermediate product. Garg et al. [45] extended this using a network similar to FlowNet [46]. However, since their network was not fully differentiable, they performed a Taylor series approximation to linearize their loss, which made the optimization difficult. To address this problem, Godard et al. [5] introduced differentiable bilinear sampling [47] into the framework of Xie et al. [43]. Additionally, they considered left-right consistency of the predicted disparities estimated from the given pair of stereo images. Their network consisted of an encoder and a decoder, where the encoder’s structure was taken from VGG-16 [48] and the decoder was composed of stacked deconvolution layers and shortcut connections from the encoder. More recently, Guo et al. [7] adopted the concept of knowledge distillation [49], and trained their monocular depth estimation network to produce the same disparity maps as the ones that were predicted by a pre-trained stereo depth estimation network.

van Dijk and de Croon [12] analyzed the behavior of monocular depth estimation and argued that depth prediction would actually depend on non-depth features such as the vertical position and the texture.

C. OUR CONTRIBUTION

We propose a method of adversarial patch attack for CNN-based monocular depth estimation methods. Similarly to some previous works [24]–[27], the patches used for our attack are recognizable to human eye. Zhang et al. [23] also attacked monocular depth estimation using imperceptibly small perturbation patterns covering the entire image, but their method was not applicable to the real world. In contrast, our patch-based approach enables physical attacks using printed patterns located in the target scene. Our method is similar to Ranjan et al.’s [27], which attacks optical flow CNNs using printed patches that are physically located in real scenes. We would like to stress several differences between our work and Ranjan et al.’s [27]. First, our method is designed to cause depth errors only around the patches rather than the entire image frame. Moreover, our method considers perspective transform [30] in the image formation model to increase the robustness of the attack in the real world. Finally, our target is monocular depth estimation, which we feel has inherent vulnerabilities due to its over-dependence on non-depth cues. To the best of our knowledge, we are the first to attack monocular depth estimation in real scenes. We also visualize the behavior of monocular depth CNNs under attack, which will contribute to a deeper understanding of monocular depth estimation in concert with other approaches (e.g., van Dijk and de Croon’s [12]).

III. PROPOSED METHOD

A. OVERVIEW

Figure 2 illustrates the overview of our method. Given a target monocular depth estimation method implemented as a CNN, our goal is to derive an adversarial patch (denoted as \( P \)) that induces the target method to produce incorrect depth estimates for the region on the input image where the patch is located.

Our method is designed to be implemented in the real world; namely, we want to deceive the target method by physically placing printed patches in the target scene. To achieve this goal, we need to consider various shooting conditions in physical settings. When a patch is printed and placed in the target scene, it is subject to a series of transformations (luminance change, geometric transformations, noise, etc.) depending on the shooting condition before it is finally
recorded in a digital image. Therefore, the imaging process of the patch is modeled so as to cover various shooting conditions.

Moreover, we implement the imaging process as fully differentiable so that the gradient with respect to the estimated depth can be propagated back to the patch through the network. During the training stage, we keep the target method’s network unchanged and update only the adversarial patch through the framework of back-propagation. We utilize the Adam optimizer [50] for this purpose. We could also utilize custom-made iterative updating rules such as FGSM [15], but we found that using the Adam optimizer led to better results.

B. MODEL OF IMAGING PROCESS

Let \( F \) be the target monocular depth estimation CNN and \( I \) the original input image. The estimated depth map is represented as \( D = F(I) \). The adversarial patch we aim to derive is denoted as \( P \). We assume that the patch \( P \) receives various transformations \( T_\theta \) through the imaging process and finally falls into the region \( R_0 \) in the input image. The attacked image is represented as \( I = I + R_0 T_\theta(P) \), where \( +R_0 \) refers to the pixel overwriting on the region \( R_0 \).

We aim to train the patch \( P \) so that the depth values in the region \( R_0 \) become a certain depth value \( d_t \), regardless of the actual depth. This is formalized as:

\[
\arg \min_P \sum_{(i,j) \in R_0} \left| d_t - F_{i,j}(I + R_0 T_\theta(P)) \right|,
\]

where \( (i,j) \) denotes a pixel coordinate in the estimated depth map. By minimizing Eq. (1), we force the estimated depth in the region \( R_0 \) to be a specific depth value \( d_t \). Depending on the value of \( d_t \), the estimated depth is guided to be different from the actual depth.

The patch \( P \) should be robust to various transformations in the imaging process. Therefore, we randomly change the transformation \( T_\theta \) for each mini-batch during training. Specifically, \( T_\theta \) includes random brightness shifts (in the \([-0.05, 0.05]\) range), contrast shifts (in the \([0.9, 1.1]\) range), addition of noise (in the \([-0.1, 0.1]\) range), scaling (with factors of \([0.1225, 0.2025]\)), and rotation (in \([-20, 20]\) degrees). Finally, the patch \( P \) receives perspective transformation, in which four vertices of the unit square \(((0, 0), (0, 1), (1, 0), (1, 1))\) are randomly perturbed as

\[
(v_x, v_y) \mapsto (v_x + u, v_y + v) \quad (u, v \in [-0.1, 0.1]),
\]

where \((v_x, v_y)\) represents each vertex of the unit square and \(u, v\) represents horizontal and vertical shifts, respectively. Only a single patch is used to cover different spatial resolutions, since we include a larger range of scaling than [27] in the transform (in [27], several patches were learned for different resolutions). We used a sufficiently large resolution for the patch \((256 \times 256)\) to reduce the unexpected effects of pixel interpolation when printing it.

C. LOSS FUNCTION

Our loss function is defined for the target adversarial patch \( P \) and is composed of three terms,

\[
L = L_{\text{depth}}(P) + \alpha L_{\text{NPS}}(P) + \beta L_{\text{TV}}(P),
\]

where \( \alpha \) and \( \beta \) are weighting coefficients that are determined experimentally.

1) DEPTH LOSS

The most important term in our loss function, the depth loss, \( L_{\text{depth}} \), is given in accordance with Eq. (1).

\[
L_{\text{depth}}(P) = \sum_{(i,j) \in R_0} \left| d_t - F_{i,j}(I + R_0 T_\theta(P)) \right|.
\]

When we attack more than one monocular depth estimation methods simultaneously, we replace Eq. (4) with the ensemble depth loss \( L_{\text{depth}}^{\text{ens}} \), as

\[
L_{\text{depth}}^{\text{ens}} = \sum_k \sum_{(i,j) \in R_0} \left| d_t - F^k_{i,j}(I + R_0 T_\theta(P)) \right|.
\]

where \( F^k \) denotes the \( k \)-th network.

2) NON-PRINTABILITY SCORE (NPS)

We included NPS [51] in our loss function to limit the color space of the patch within the printable color gamut, as

\[
L_{\text{NPS}}(P) = \sum_{i,j} \min_{\bar{C} \in C} \| \bar{p}_{i,j} - \bar{c} \|_1,
\]

where \( \| \cdot \|_1 \) denotes the \( \ell_1 \) norm and \( p_{i,j} \) denotes the color vector of a pixel \((i, j)\) in the patch \( P \). We seek the closest color vector \( c \) from the set of printable colors \( C \). A smaller \( L_{\text{NPS}} \) means better printability.

3) TOTAL VARIATION (TV)

Non-smooth patches are more likely to be affected by aliasing artifacts when they are printed and captured by a camera. Therefore, we encourage the smoothness of the patch \( P \) by using total variation (TV) loss, similarly to [52].

\[
L_{\text{TV}}(P) = \sum_{i,j} \sqrt{(p_{i,j} - p_{i,j+1})^2 + (p_{i,j} - p_{i,j+1})^2}.
\]

D. IMPLEMENTATION

1) TARGET METHODS

As the target of our proposed attack, we used two state-of-the-art monocular depth estimation methods [7], [8].

Guo et al.’s method [7] is trained in an unsupervised manner using knowledge distillation. Their network consists of an encoder and a decoder. The encoder part is implemented using VGG-16 [48] and the decoder part is composed of stacked deconvolution layers and skip connections. The output from the network is a disparity map, which is converted into the depth map using the relation:

\[
\text{disparity} = \text{(baseline} \times \text{focal length)}/\text{depth},
\]
where the baseline and the focal length were provided in the KITTI dataset.

Meanwhile, Lee et al.’s [8] takes a supervised training framework, where the estimated depth is directly supervised by the corresponding ground-truth. Their network consists of an encoder and a decoder as well, but the structures are more complex than Guo et al.’s [7]; DenseNet-161 [53] was adopted as the encoder, and atrous spatial pyramid pooling [54] and local planar guidance layers were used in the decoder. This method is one of the top performing monocular depth estimation methods in the KITTI benchmark [55].

These two networks were pre-trained on the KITTI dataset [56] using a data-splitting rule proposed by Eigen et al. [32]. The Eigen split consisted of 22,600 stereo image pairs for training, 888 for validation, and 697 for testing.

2) TRAINING DETAILS
We trained adversarial patches under several conditions. We tested two target depths ($d_t = 3$ m and 150 m) and three target configurations (individual and simultaneous attacks to either and both of [7] and [8]).

For training of our adversarial patch, we used only the left view images from the Eigen’s 22,600 training split [32]. The width and height of the input images were set to 512 and 256 pixels, respectively to fit the input size of the target networks. The input images were randomly augmented by horizontal flipping, zooming with a factor of [0.8, 1.2], gamma correction in the range of [0.8, 1.2], illumination change with a factor of [0.8, 1.2], and color jittering in the range of [0.95, 1.05]. The resolution for the patch was set to $256 \times 256$ pixels, but its apparent size in the input image was changed by the patch transformer $T_{\theta}$.

We used a Linux-based PC equipped with a NVIDIA GeForce GTX 1080 Ti. The networks were implemented using Python version 3.6.9 and PyTorch [57] version 1.1.0. We used the Adam optimizer [50] with learning rate $10^{-3}$, and batch size was set to 8. The number of epochs was 40.

The resulting adversarial patches are shown in Fig. 4.

![Figure 3. Results of digital adversarial attacks.](image)

![Figure 4. Adversarial patches trained with various conditions.](image)
FIGURE 5. Results of digital adversarial attacks by $P^1_n$ and $P^1_f$ on Guo et al.’s method [7].

As shown in (a), our attack was quite effective for Guo et al.’s method [7]: in both cases, namely, where the patch was trained exclusively for Guo et al.’s method [7] and where it was trained for both the methods [7], [8], the network produced incorrect depths corresponding to $d_i$ in the regions where the patches were located.

In contrast, as shown in Fig. 3(b), the attack had a limited effect for Lee et al.’s method [8]: in particular, the patches trained for both methods were less effective. One possible
reason for this is the higher complexity of Lee et al.’s method [8] compared to Guo et al.’s [7], which would bring more robustness to attacks. However, we conclude that our attack was effective to some extent because significant depth errors were induced by the adversarial patches.

It should be noted that simultaneous attack was possible for both methods, which had different network structures and training strategies. See the results with $P_n^1$ and $P_t^1$ in Fig. 3. This would indicate the inherent vulnerability of monocular depth estimation, where few geometric cues are available from the input image itself.

To further demonstrate the effectiveness of our attack, we present additional results on Guo et al.’s method [7] in Fig. 5, which shows that our adversarial patches were effective in various conditions (location, scale, orientation, and background).

**B. PHYSICAL ADVERSARIAL ATTACK**

We also conducted a physical adversarial attack by using the printed adversarial patches. Figure 6 shows several input images and estimated depth maps. We can see a similar tendency as the one with the digital adversarial attack: namely, the attack was less effective against Lee et al.’s method [8] than Guo et al.’s [7]. In particular, the effect of $P_t^2$ on Lee et al.’s method [8] was almost non-existent. This is seemingly related to the property of Lee et al.’s method that the estimated depth is closely related to the vertical position in the image; $P_t^2$ ($d_t = 150$ m) was placed near the bottom of the image, where Lee et al.’s method is more likely to produce small depth values. In contrast, when $P_t^2$ was placed near the top of the image, we obtained a result closer to our intention (see Fig. 8). To conclude, both of the monocular depth estimation methods could be attacked in the real world, although the results were not always sufficient against Lee et al.’s method.

**FIGURE 6. Results of physical adversarial attacks.**

**FIGURE 7. Results of physical adversarial attacks on Guo et al.’s method [7] under various conditions.**

In Fig. 7, we present more results of the physical adversarial attack against Guo et al.’s method. In (b), the patch $P_t^1$ was located upright, but in (c) and (d), it was rotated

**FIGURE 8. Successful case of our attack on Lee et al.’s method [8].**
by ±15 degrees. The attack was effective for all three cases, indicating the robustness of the adversarial patch derived by our method. In (e), we can see that the patch $P^1_f$ (trained for Lee et al.’s method) had little effect on Guo et al.’s method, although the appearance of $P^*_{f}$ was similar to that of $P^1_{f}$, and in (f), $P^*_{f}$ was effective on the same method. This indicates that similarity to the human eye does not always correspond to the similarities to the depth estimation methods.

Please refer to the supplementary video for more results.

V. ANALYSIS THROUGH VISUALIZATION

As discussed in the previous section, the effect of our adversarial attack was spatially localized in the resulting depth map; incorrect depth estimates were induced only in the regions around the adversarial patches. To analyze this effect, we present two methods for visualizing the effects of adversarial patches. First, we visualize the potential regions on a depth map where the depth values are likely to be affected by an adversarial patch. Second, we visualize the network’s activation incurred by the adversarial patch, which helps to analyze the mechanism where the adversarial patch induces incorrect depth estimates. We use digital attack (adversarial patches superimposed by manipulating the image digitally) for this analysis, similarly to the work of van Dijk and de Croon [12]. We adopted Guo et al.’s method [7] as the target of visualization.

A. POTENTIALLY AFFECTED REGIONS

Given a region $R_\theta$ for an adversarial patch to be placed in the input image $I$, we can predict the potential effect on the resulting depth value $D(u, v)$ for a pixel $(u, v)$ as

$$H(u, v) = \left| \frac{\partial D(u, v)}{\partial I(i, j, c)} \right|,$$

where $D = F(I)$. In the above equation, the partial derivative can be obtained by standard back-propagation. The map $H(u, v)$ shows how much the estimated depth of each pixel $(u, v)$ is potentially affected by the adversarial patch located at $R_\theta$. Note here that we only consider the location $R_\theta$, and do not specify the pattern for the adversarial patch. Therefore, with this visualization, we actually analyze the intrinsic sensitivity of the pre-trained target network $F$.

Figure 9 shows an example of this visualization. The original input image and an attacked image are shown in (a), (b), and (c) from which Guo et al.’s method [7] predicted the depth maps shown in (d), (e), and (f), respectively. Shown in (g) is $H(u, v)$, a prediction of the potentially affected region, where $R_\theta$ was set to the region of the adversarial patch in (b) and (c). To verify this prediction, the difference between (d) and (e) is presented in (h) and the difference between (d) and (f) is presented in (i). As we can see, the predicted region in (g) was well aligned to the resulting depth differences in (h) and (i).
B. NETWORK ACTIVATION

For classification tasks, a well-known method for visualizing network activation is Grad-CAM [58]. This method can be applied only for classification tasks, where the output from the network is typically a global label and no spatial information is involved. However, in our case, the output from the network is a pixel-wise depth map that depends on the location of a pixel. Therefore, we extend the idea of Grad-CAM to our problem as follows.

First, we focus on the encoder part of the target network $F$. We obtain a layer-wise activation map for a spatial position $(u, v)$ in the output depth map $D(u, v)$ as

$$G_{m}^{u,v}(i,j) = \sum_{k} \frac{\partial D(u, v)}{\partial A_{km}^{m}(i,j)} \cdot A_{km}^{m}(i,j),$$

(10)

where $A_{km}^{m}(i,j)$ is the $k$-th feature map of the $m$-th convolution layer and $(i,j)$ denotes the spatial location on the feature map. We then resize activation maps $G_{m}^{u,v}(i,j)$ to the same size as the input image by bilinear interpolation and aggregate them over all the encoder’s layers. We finally take the summation over the regions $(u, v) \in R_{\theta}$ to obtain the final activation map, as

$$G(i,j) = \sum_{(u,v) \in R_{\theta}} \left( \sum_{m} \text{Resize}(G_{m}^{u,v}(i,j)) \right).$$

(11)

This activation map shows the extent to which spatial neighbors are involved for the erroneous estimates on $(u, v) \in R_{\theta}$, when an adversarial patch covers the region $R_{\theta}$.

VI. CONCLUSION

We have proposed a method of adversarial patch attack for CNN-based monocular depth estimation methods. The adversarial patches were trained to induce incorrect depth estimation around the region where they were located, in the presence of various transformations such as perspective transform, scaling, and translation. We demonstrated that our method can be implemented in the real world by placing a printed pattern in the target scene. We also analyzed the behavior of monocular depth methods under attack by visualizing the potentially affected regions and activation maps. To the best of our knowledge, we are the first to achieve physical adversarial attacks on depth estimation methods. Our future work will include extending our method to similar tasks such as optical flow estimation and stereo depth estimation. We hope our work will lead to a wider recognition of the vulnerability of monocular depth estimation methods and thus to the development of safer depth estimation techniques.

REFERENCES

[1] J. L. Schonberger and J.-M. Frahm, “Structure-from-Motion revisited,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 4104–4113.
KOICHIRO YAMANAKA received the B.E. degree in electrical engineering from Nagoya University, Japan, in 2019, where he is currently a Graduate Student with the Graduate School of Engineering. His research interests include monocular depth estimation and adversarial examples.

RYUTAROH MATSUMOTO (Member, IEEE) was born in Nagoya, Japan, in November 29, 1973. He received the B.E. degree in computer science, the M.E. degree in information processing, and the Ph.D. degree in electrical and electronic engineering from the Tokyo Institute of Technology, Japan, in 1996, 1998, and 2001, respectively. He was an Assistant Professor, from 2001 to 2004, and an Associate Professor, from 2004 to 2017 with the Department of Information and Communications Engineering, Tokyo Institute of Technology. He also served as a Velux Visiting Professor with the Department of Mathematical Sciences, Aalborg University, Denmark, in 2011 and 2014. From 2017 to 2020, he was an Associate Professor with the Department of Information and Communication Engineering, Nagoya University, Japan, under the 6-University Human Assets Promotion Program for Innovative Education and Research Program (6-U HAPPIER). He returned to the Tokyo Institute of Technology after the expiration of the 6-U HAPPIER. His research interests include error-correcting codes, quantum information theory, information theoretic security, and communication theory. He received the Young Engineer Award from IEICE and the Ericsson Young Scientist Award from Ericsson Japan, in 2001. He received the best paper awards from IEICE in 2001, 2008, 2011, and 2014.

KEITA TAKAHASHI (Member, IEEE) received the B.E., M.S., and Ph.D. degrees in information and communication engineering from The University of Tokyo, in 2001, 2003, and 2006, respectively. He was a Project Assistant Professor with The University of Tokyo, from 2006 to 2011, and an Assistant Professor with The University of Electro-Communications, from 2011 to 2013. He is currently an Associate Professor with the Graduate School of Engineering, Nagoya University, Japan. His research interests include computational photography, image-based rendering, and 3-D displays.

TOSHIKAI FUJII (Member, IEEE) received the B.E., M.E., and Dr.E. degrees in electrical engineering from The University of Tokyo, in 1990, 1992, and 1995, respectively. Since 1995, he has been with the Graduate School of Engineering, Nagoya University, where he is currently a Professor. From 2008 to 2010, he was with the Graduate School of Science and Engineering, Tokyo Institute of Technology. His research interests include multidimensional signal processing, multicamera systems, multiview video coding and transmission, free-viewpoint television, and related applications.