Pattern Recognition Letters 167 (2023) 82–89

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

# Local Selective Vision Transformer for Depth Estimation Using a Compound Eye Camera

Wooseok Oh, Hwiyeon Yoo, Taeoh Ha, Songhwai Oh*

*Department of Electrical and Computer Engineering, ASRI, Seoul National University, Seoul 08826, South Korea*

ARTICLE INFO

*Article history:*
Received 25 July 2022
Revised 31 January 2023
Accepted 5 February 2023
Available online 7 February 2023

Edited by Jiwen Lu

*Keywords:*
Compound Eye
Depth Estimation
Vision Transformer

ABSTRACT

A compound eye camera is a hemispherical camera made by mimicking the structure of an insect's eye. In general, a compound eye camera is composed of a set of single eye cameras. The compound eye camera has various advantages due to its unique structure and can be used in various vision tasks. In order to apply the compound eye camera to various vision tasks using 3D information, depth estimation is required. However, due to the difference between the compound eye image and the 2D RGB image, it is hard to use the existing depth estimation methods directly. In this paper, we propose a transformer-based neural network for eye-wise depth estimation, which is suitable for the compound eye image. We modify the self-attention module with local selective self-attention to take advantage of the compound eye's hemispherical structure. In addition, we reduce the computational amount and increase the performance through the eye selection module. Using the proposed local selective self-attention and eye selection modules, we are able to improve the performance without large-scale pre-training. Compared to the ResNet-based depth estimation network, our method showed 2.8% and 1.4% higher performance on the GAZEBO and Matterport3D datasets, respectively, with 15.3% fewer network parameters.

## 1. Introduction

The compound eye camera, modeled after an insect's eye, is a set of small resolution single eye cameras regularly distributed in a hemispherical shape. An example of a compound eye camera image can be seen in Fig. 1. The compound eye camera has various advantages, such as low aberration and large field of view (FOV) due to the characteristics of this structure [1]. There have been studies to develop the compound eye camera hardware in real world [2]. Most of these compound eye camera hardwares have low-resolution observations and have been applied on tasks such as medical endoscopy [3].

On the other hand, there have been studies that focus on the advantage of the compound eyes and apply them to various vision tasks [4–7]. Among vision tasks, depth information is critical in tasks such as 3D reconstruction [8,9] and mobile robot navigation [10]. In a compound eye camera, a depth sensor must be attached to every single eye to obtain depth information using a depth sensor. However, since depth sensors are relatively expensive, it is costly to obtain depth information for each eye in a compound eye camera. Therefore, estimating depth from RGB images can be a feasible alternative solution to obtain depth information.

Depth estimation has been studied in various ways. As deep learning technology develops in computer vision, the study of applying deep neural networks to depth estimation has became the mainstream. Specifically, convolutional neural network (CNN) based depth estimation techniques have been studied extensively [7,11,12].

These days, the transformer structure has been applied to depth estimation, improving the performance [13,14]. The transformer [15] is a structure that extracts features using a self-attention module that finds out how much attention should be paid to each input token. By using a patch of an image as an input token [16], the transformer structure has been applied to various computer vision tasks.

Among these various depth estimation studies, [7] performed depth estimation on a compound eye image, not a 2D RGB image. [7] proposed an eye-wise depth estimation method that predicts one depth value for each single eye of a compound eye image. Eye-wise depth estimation is suitable for compound eye cameras mainly used in mobile robots because of the advantages of mem-

[*] Corresponding author.
*E-mail addresses:* wooseok.oh@rllab.snu.ac.kr (W. Oh), hwiyeon.yoo@rllab.snu.ac.kr (H. Yoo), timothy.ha@rllab.snu.ac.kr (T. Ha), songhwai@snu.ac.kr (S. Oh).

https://doi.org/10.1016/j.patrec.2023.02.010
0167-8655/© 2023 Elsevier B.V. All rights reserved.

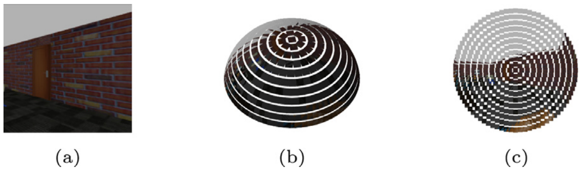

**Fig. 1.** An illustration of a 2D RGB image and compound eye images taken at the same location in a GAZEBO simulator. (a) An example of a 2D RGB image. (b) An example of a compound eye image. (c) An example of a compound eye image projected in 2D.

ory and computation. However, the 2D CNN-based network used in [7] cannot fully utilize the 3D hemispherical structure of the compound eye, so there is a room for improvement.

In this paper, we propose a method for eye-wise depth estimation using a transformer structure suitable for compound eye images. Since a single eye image can be considered as a patch of a vision transformer, the structure of the compound eye image itself is suitable to use the transformer. However, due to the difference of the data structures, the conventional vision transformer methods designed for the 2D images cannot be used for the compound eye images directly. In addition, large-scale pre-training is required to obtain good performance in a transformer-based network. Unfortunately, since there is no ideal compound eye camera hardware with high resolution and curvature, there is no large dataset, so large-scale pre-training is not possible.

To solve these problems, we propose a local selective self-attention module that applies attention to the closest $k$ single eyes by measuring the physical distance between the single eye cameras. The proposed module is suitable for the compound eye because it can utilize locality in a hemispherical structure. In addition, to reduce the computational cost and increase the performance, we propose an eye selection module that selectively uses a subset of the adjacent eyes in the attention module rather than using all of them. To the best of our knowledge, our method is the first to propose depth estimation using a transformer structure in a compound eye image. The proposed network performed eye-wise depth estimation and obtained 2.8% and 1.4% higher accuracy with 15.3% fewer network parameters than the CNN-based network [7] on two datasets without large-scale pre-training.

Our main contributions are as follows:

- We suggest a network design that utilizes the hemispherical structure of the compound eye.
- To the best of our knowledge, this is the first application of a transformer structure to compound eye depth estimation.

## 2. Related Work

### 2.1. Application of Deep Neural Networks to Compound Eye Images

There are studies dealing with various computer vision tasks using compound eye images, and recently, deep neural networks, especially CNNs, are being applied to compound eye images. For example, [4] proposed a network for estimating objectness on a compound eye image, [5] proposed a low complexity semantic segmentation scheme based on a CNN, and [6] proposed a neural network to classify the ego-motion of a compound eye camera. However, since there is no compound eye image data, the neural networks of [4–6] were learned from the images obtained by converting 2D RGB images of the real-world into compound eye images. On the other hand, [7] collected a dataset from simulation using a compound eye camera, proposed a deep neural network that performs depth estimation using the collected dataset, and proposed a simple method for 3D reconstruction using the estimated depth values.

Unlike previous studies, we perform depth estimation using a transformer-based network. Also, we train the network from the compound eye image dataset collected from simulation and the photorealistic compound eye image dataset converted from real-world 2D RGB images.

### 2.2. Vision Transformer

Since the success of Vision Transformer (ViT) [16], there have been many studies trying to use a transformer as a backbone in computer vision. Among them, [17–19] proposed a network suitable for dense prediction to extract multi-resolution features using the transformer architecture. [17,18] proposed a method for extracting multi-resolution features through a hierarchical structure and applied it to image classification, object detection, and semantic segmentation. On the other hand, [19] proposed a method of extracting multi-resolution features in a parallel manner and applied it to image classification, semantic segmentation, and human pose estimation.

ViT has the disadvantage of a high computational cost. In addition, ViT requires a large amount of pre-training and dataset compared to CNN-based models. In [18,19], the amount of computation is reduced by dividing the input image into windows and applying self-attention only within each window. [19–21] uses convolution and transformer together to make learning more robust by adding an inductive bias to the model through the locality of the convolution. In this way, they showed good performance even with a relatively small amount of dataset.

Among various methods, the methods most relevant to ours are [21] and [22]. [21] proposes a self-attention module that applies attention to adjacent pixels in the 2D image, and [22] proposes a self-attention module that obtains an attention map from image patches with $k$ keys similar to a query. Unlike previous methods, the proposed method is designed for a compound eye structure and applies attention to physically adjacent eye units.

### 2.3. Monocular Depth Estimation

Most of the monocular depth estimation algorithms using deep neural networks are CNN-based methods [11,12]. The structure of the network and the loss function significantly affect the performance of depth estimation. [11] used two networks, one for coarse estimation and the other for refining coarsely estimated values. They also used a scale-invariant loss independent of the global scale of the depth value during training to solve the problem of being sensitive to the scale of the data. [12] proposed a network that performs depth estimation using multi-resolution features and used three complementary losses for training to obtain clear boundaries.

Recently, there have been studies that applied the transformer to depth estimation. [13,14] propose methods for monocular depth estimation using the transformer architecture. Among them, [13] uses the feature selected from ResNet [23] as an input of the transformer network and performs depth estimation by fusing multi-resolution features in parallel.

## 3. Background

### 3.1. Compound Eye Camera Structure

A compound eye camera has a hemispherical structure in which single eye cameras are arranged in circular layers (Fig. 2a). Here, the layer is a set of single eyes with the same elevation angle. Each single eye camera is a camera with low resolution, and in the $n$th layer, $m_n$ single eye cameras are circularly located at regu-

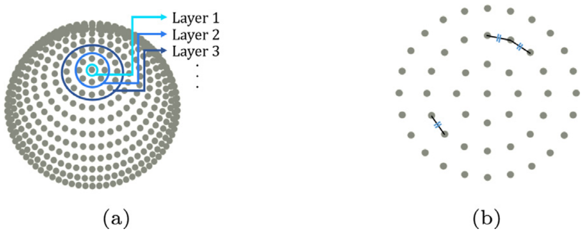

**Fig. 2.** An illustration of the compound eye camera. Each gray dot represents a single eye of the compound eye. (a) A compound eye camera with 11 layers. (b) Top view of the compound eye camera with 4 layers.

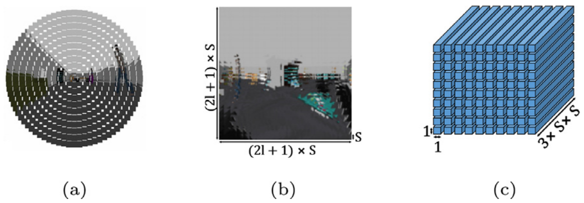

**Fig. 3.** An illustration of the compound eye images in various forms. All three examples used compound eye images with 11 layers. (a) Example of visualization by projecting a compound eye image (b) Example of visualizing a compound eye image in the same form as a 2D RGB image (c) An example of vectorizing a compound eye image.

lar intervals (Fig. 2b). The elevation angle between adjacent layers is identical for all layers.

### 3.2. Compound Eye Data for Neural Network

There have been studies that applied neural networks to compound eye images [4–7]. They used the compound eye image data format suitable for the neural network, and a similar format is used in this paper. To arrange the compound eye image (Fig. 3a) in a rectangular shape like a 2D RGB image, they use a compound eye camera with a structure in which only one single eye exists at the center of the camera and $8n - 8$ single eyes in the $n$th layer. In this configuration, a compound eye image can be visualized in a 2D RGB image form. An example of a compound eye image in the form of 2D RGB image is illustrated in Fig. 3b. In order to make it suitable for the neural network's input, we use a compound eye image vectorized as a tensor. The vectorized compound eye image has a representation of $\mathbb{R}^{(2l-1)\times(2l-1)\times 3S^2}$ [4], where $l$ is the number of layers and $S \times S$ is the resolution of a single eye. The structure of vectorized compound eye image can be seen in Fig. 3c. The compound eye depth image has one value for each single eye, so it has a representation of $\mathbb{R}^{(2l-1)\times(2l-1)\times 1}$ [7].

## 4. Method

We propose a transformer-based network for eye-wise depth estimation which estimates one depth value per single eye. We use a transformer block and a convolution layer together. The network estimates one depth value for each single eye unit using a vectorized compound eye RGB image as network input. An overview of our network can be seen in Fig. 4.

We propose local selective self-attention, described in Section 4.1, that adds locality to the transformer network and propose an eye selection module, described in Section 4.2, that reduces the computational complexity and improves performance. Through these modules suitable for the compound eye structure, it became possible to reduce the amount of computation and increase performance on a small dataset without pre-training.

### 4.1. Local Selective Multi-Head Self Attention

Most vision transformer networks are pre-trained on large datasets to improve performance. However, large-scale pre-training on compound eye image data is difficult due to the lack of publicly available large datasets. In order to obtain good performance with the transformer structure without pre-training, it is necessary to add an inductive bias to the model, such as using a CNN with the transformer. To add locality to the transformer structure, we modify the self-attention module to obtain attention maps in the local area from the compound eye image. As shown in Fig. 5, adjacent eyes in the compound eye image are different from those in the vectorized data format. Therefore, we modified the self-attention structure to extract features by applying attention between the eyes that are physically close to each other on the compound eye camera.

In ViT, image patches are used as input to the network. Given an image $x \in \mathbb{R}^{h\times w\times d}$ of height $h$, width $w$ and channel dimension $d$, the image can be cropped into image patches of size $p \times p$. Let the set of image patches be $X \in \mathbb{R}^{\frac{h}{p}\times\frac{w}{p}\times D}$, and let a patch of position at $ij$ be $X_{ij} \in \mathbb{R}^D$, where $D$ is $d \times p \times p$. Then ViT's self-attention formulation for each attention head is as follows:

$$f_{ij}^m = \sum_{a,b\in\mathbf{N_p}} \text{softmax}_{ab}\left(\frac{q_{ij}^{m^T} k_{ab}^m}{\sqrt{D/M_h}}\right) v_{ab}^m, \tag{1}$$

where $\mathbf{N_p} = \{a, b \mid 0 \le a < \frac{h}{p}, 0 \le b < \frac{w}{p}\}$, $M_h$ is the number of attention heads and $f_{ij}^m$ is the output of the $m$th attention head. In this formulation, the queries $q_{ij}^m$, keys $k_{ab}^m$, and values $v_{ab}^m$ are linear transformation of the fraction of patches $X_{ij}^m$ and $X_{ab}^m$, where $X_{ij}^m$ and $X_{ab}^m$ have a representation of $\mathbb{R}^{D/M_h}$. The output of the self-attention module is as follows:

$$f_{ij} = \text{Concat}[f_{ij}^1, \cdots, f_{ij}^{M_h}]. \tag{2}$$

Then, the formulation of the local selective self-attention for each attention head is as follows:

$$f_{ni}^m = \sum_{a,b\in\mathbf{N_k}(n,i)} \text{softmax}_{ab}\left(\frac{q_{ni}^{m^T} k_{ab}^m}{\sqrt{D/M_h}}\right) v_{ab}^m, \tag{3}$$

where $\mathbf{N_k}(n, i)$ is a set of indices of $k$ single eyes with large cosine similarity to the direction vector of the $i$th single eye of the $n$th layer. We use a single eye image as an image patch in local selective self-attention.

The local selective self-attention can control locality as the value of $k$ changes, and if $k$ is small, it takes a small amount of computation compared to other attention methods.

### 4.2. Eye Selection

The eye selection method is a sparse attention method with an attention map obtained from the selected eyes. The attention coefficient is used to determine which eyes are paid attention to. The attention coefficient is the ratio of reflecting each eye's image to extracting the eye's features in a specific location. It is constant for the input image, unlike the attention value, which is dependent on the input image. Through this attention coefficient, we can select the eyes that are important for extracting features regardless of the input image. We then use the attention value and attention coefficient to find the degree to which those eyes are used to extract features. For each eye of each head, $n_e$ eyes with the largest attention coefficient are selected.

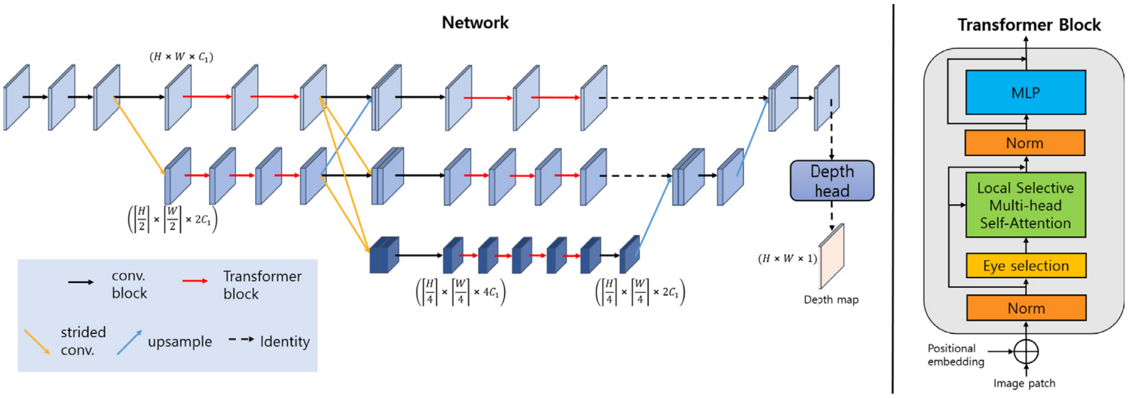

**Fig. 4.** Overview of our depth estimation network. Our network fuses parallel multi-resolution features extracted from transformer blocks and estimates the depth through a simple depth head. In our transformer block, instead of multi-head self-attention of ViT, we use the eye selection module and local selective multi-head self-attention module.

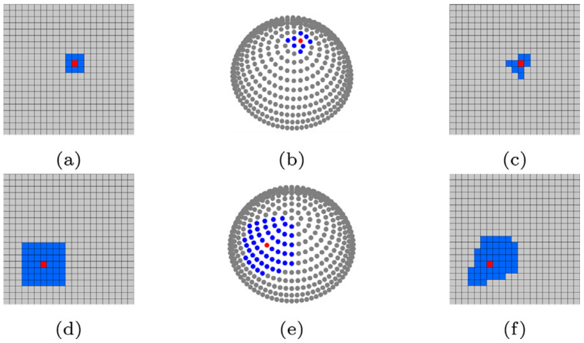

**Fig. 5.** An illustration of adjacent pixels on a 2D RGB image and adjacent eyes on a compound eye image. (a) and (d) are 21x21 2D images and (b) and (e) represents compound eye with 11 layers and 441 single eyes. (c) and (f) show vectorized form of (b) and (e). Based on the red pixel, adjacent pixels are expressed in blue. (a)∼(c) represent eight adjacent pixels, and (d)∼(f) represent 48 adjacent pixels.

For each attention head, the formula applying the eye selection method to local selective self-attention is as follows:

$$f_{ij}^{m} = \sum_{a,b\in \mathbf{N_s}(i,j)} \mathrm{Norm}_{ab}\left(\mathrm{softmax}_{ab}\left(\frac{q_{ij}^{m\,T} k_{ab}^{m}}{\sqrt{D/M_h}}\right) c_{ijab}^{m}\right) v_{ab}^{m}, \tag{4}$$

where $Norm_{ab}$ is a normalization such that the sum of the values is 1, $c_{ijab}^{m} \in [0,1]$ is the attention coefficient and $\mathbf{N_s}(i,j)$, a subset of $\mathbf{N_k}(i,j)$, is a set of indices of $n_e$ single eyes with the largest attention coefficient.

The attention coefficient is a learnable parameter, and its value is updated during learning. Since the selected eyes are changed during training, a similar effect to dropout [24] can be obtained.

### 4.3. Training Details

Our network structure can be seen in Fig. 4. HRFormer [19] is modified and used as the backbone network. A simple depth head is used as a decoder to estimate the depth value from the features extracted from the backbone. The depth head consists of two linear layers whose activation is a ReLU function. Each has 2, 4, and 8 attention heads from high-resolution transformer blocks to low-resolution transformer blocks. We use the learnable positional embedding used in ViT.

We use the log mean absolute error (log-mae) as the training loss $L$ to train the depth estimation network.

$$L = \frac{1}{M_e}\sum_{i=1}^{M_e} |\log(d_i) - \log(g_i)|, \tag{5}$$

where $d_i$ is the estimated depth value of the $i$th single eye, $g_i$ is the ground truth depth value of $i$th single eye and $M_e$ is the number of single eyes. We use the Adam optimizer and set the initial learning rate as 0.001, weight decay as 0.0001, epoch as 60, and $C_1$ as 64.

## 5. Experimental Results

In this section, we evaluate the performance of our proposed network and comparing it to other methods: different backbone networks trained from scratch on compound eye images and transformer-based depth estimation methods trained on 2D images. We also investigate the effect of using existing pre-trained network for 2D images on depth estimation in compound eye. In addition, we find out how the local selective self-attention and eye selection methods affect the performance and computational complexity. Lastly, we analyze the advantages of eye-wise depth estimation.

### 5.1. Dataset

We learn the depth estimation network from two datasets: GAZEBO simulation [25] dataset and Matterport3D dataset [26]. The GAZEBO simulation dataset is collected using a compound eye with 11 layers and 10 × 10 resolution in the simulation in the same way as [7]. We train a depth estimation network using 7,200 images and test it on other 4,000 images. Since there is no high-resolution images collected with the compound eye in the real world, we use a dataset that transformed the image of the Matterport3D dataset into a compound eye image with 11 layers and a resolution of 10 × 10. We train a depth estimation network using 32,540 images and test it on other 11,976 images. We use only regions with ground truth depth values within 4.5 m.

### 5.2. Evaluation Metrics

We compare our method with other existing methods using several metrics. In addition to the log mean absolute error (log-mae) described in Section 4.3, the following metrics are used to evaluate the performance of several depth estimation methods.

- root mean squared error (rms): $\sqrt{\frac{1}{M_e}\sum_{i=1}^{M_e}(d_i - g_i)^2}$
- mean relative error (rel): $\frac{1}{M_e}\sum_{i=1}^{M_e}\frac{|d_i - g_i|}{g_i}$
- mean absolute error (mae): $\frac{1}{M_e}\sum_{i=1}^{M_e} | d_i - g_i |$
- threshold accuracy: $\%\mathrm{of} d_i \mathrm{s.t} \max(\frac{d_i}{g_i}, \frac{g_i}{d_i}) = \delta < \delta_{thr}$ for $\delta_{thr} = 1.25, 1.25^2, 1.25^3$

**Table 1**
Comparison of different methods on the GAZEBO dataset. $C_1$ is the feature dimension of the first stage of transformer block.

| Method | | Accuracy | | | Error | | | | Params | FLOPs | Time |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ | log-mae | mae | rel | rms | (M) | (G) | (ms/it) |
| **ConvolutionalNetworks** | ResNet18 [7] | 0.7257 | 0.8401 | 0.9016 | 0.3451 | 0.2261 | 0.3472 | 0.6007 | 13.2540 | **0.2753** | 5.8553 |
| | ResNet50 [7] | 0.7160 | 0.8265 | 0.8881 | 0.3867 | 0.2469 | 0.4170 | 0.6553 | 49.3380 | 0.6370 | 9.3107 |
| **Transformers** | ViT-Base($C_1 = 64$) [16] | 0.5915 | 0.7800 | 0.8629 | 0.5401 | 0.3348 | 0.6307 | 0.7854 | **0.6081** | 0.5528 | **2.6307** |
| | ViT-Base($C_1 = 128$) [16] | 0.5944 | 0.7843 | 0.8663 | 0.5246 | 0.3300 | 0.5965 | 0.7592 | 2.3140 | 1.5910 | 2.6461 |
| | ViT-Base($C_1 = 256$) [16] | 0.6363 | 0.7962 | 0.8663 | 0.5018 | 0.3175 | 0.6192 | 0.7520 | 9.0180 | 5.1220 | 2.6752 |
| | Swin-T($C_1 = 64$) [18] | 0.6742 | 0.7905 | 0.8551 | 0.4807 | 0.3011 | 0.6309 | 0.7931 | 14.0300 | 0.4588 | 6.3447 |
| | Swin-T($C_1 = 128$) [18] | 0.6909 | 0.8122 | 0.8770 | 0.4216 | 0.2690 | 0.5154 | 0.6906 | 55.9690 | 1.7910 | 6.4356 |
| **Transformerswith Conv.** | CvT-13($C_1 = 64$) | 0.6972 | 0.8410 | 0.9068 | 0.3559 | 0.2298 | 0.3250 | 0.5981 | 20.5300 | 0.9216 | 10.1826 |
| | HRFormer-S($C_1 = 32$) [19] | 0.7211 | 0.8374 | 0.8944 | 0.3853 | 0.2428 | 0.4534 | 0.6712 | 7.5750 | 0.3815 | 23.9575 |
| | HRFormer-S($C_1 = 64$) [19] | 0.7319 | 0.8441 | 0.9049 | 0.3584 | 0.2249 | 0.3863 | 0.6391 | 30.0010 | 1.4640 | 24.3935 |
| | **Ours**($C_1 = 64$) | **0.7536** | **0.8606** | **0.9168** | **0.3068** | **0.2026** | **0.2875** | **0.5635** | 11.2270 | 0.7687 | 11.2067 |

**Table 2**
Comparison of different methods on the Matterport3D dataset. $C_1$ is the feature dimension of the first stage of transformer block.

| Method | | Accuracy | | | Error | | | | Params | FLOPs | Time |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ | log-mae | mae | rel | rms | (M) | (G) | (ms/it) |
| **ConvolutionalNetworks** | ResNet18 [7] | 0.5055 | 0.7750 | 0.9032 | 0.5411 | 0.2967 | 0.3208 | 0.7445 | 13.2540 | **0.2753** | 5.8553 |
| | ResNet50 [7] | 0.4930 | 0.7642 | 0.8971 | 0.5569 | 0.3046 | 0.3296 | 0.7624 | 49.3380 | 0.6370 | 9.3107 |
| **Transformers** | ViT-Base($C_1 = 64$) [16] | 0.3232 | 0.6106 | 0.8149 | 0.7640 | 0.4119 | 0.4827 | 0.9589 | **0.6081** | 0.5528 | **2.6307** |
| | ViT-Base($C_1 = 128$) [16] | 0.3265 | 0.6145 | 0.8165 | 0.7592 | 0.4099 | 0.4916 | 0.9478 | 2.3140 | 1.5910 | 2.6461 |
| | ViT-Base($C_1 = 256$) [16] | 0.3333 | 0.6230 | 0.8216 | 0.7492 | 0.4038 | 0.4715 | 0.9456 | 9.0180 | 5.1220 | 2.6752 |
| | Swin-T($C_1 = 64$) [18] | 0.4486 | 0.7381 | 0.8839 | 0.6041 | 0.3266 | 0.3838 | 0.7911 | 14.0300 | 0.4588 | 6.3447 |
| | Swin-T($C_1 = 128$) [18] | 0.4366 | 0.7298 | 0.8813 | 0.6136 | 0.3319 | 0.3795 | 0.8052 | 55.969 | 1.7910 | 6.4356 |
| **Transformerswith Conv.** | CvT-13($C_1 = 64$) | 0.4749 | 0.7490 | 0.8883 | 0.5820 | 0.3165 | 0.3483 | 0.7943 | 20.5300 | 0.9216 | 10.1826 |
| | HRFormer-S($C_1 = 32$) [19] | 0.4645 | 0.7555 | 0.8966 | 0.5808 | 0.3137 | 0.3566 | 0.7712 | 7.5750 | 0.3815 | 23.9575 |
| | HRFormer-S($C_1 = 64$) [19] | 0.4805 | 0.7668 | 0.9027 | 0.5631 | 0.3049 | 0.3372 | 0.7598 | 30.0010 | 1.4640 | 24.3935 |
| | **Ours**($C_1 = 64$) | **0.5190** | **0.7885** | **0.9115** | **0.5214** | **0.2867** | **0.3094** | **0.7201** | 11.2270 | 0.7687 | 11.2067 |

### 5.3. Performance Comparison

#### 5.3.1. Comparison to various backbone networks

We set the $k$ to 49 and the number of selected eyes in the eye selection module to nine. We compare our network with baseline eye-wise depth estimation networks using three types of backbone networks on compound eye image datasets such as GAZEBO and Matterport3D dataset. The first type is CNN-based networks such as ResNet18 and ResNet50 [23], the second is networks using only transformer structures such as ViT-Base [16] and Swin-T [18], and the last is networks using transformer and convolution together such as CvT-13 [20] and HRFormer-S [19]. The comparison results can be seen in Tables 1 and 2. The depth images estimated by our network and the ground truth depth images can be seen in Figs. 6 and 7. In addition, examples of images where our method incorrectly estimates depth and the results are shown in Fig. 8.

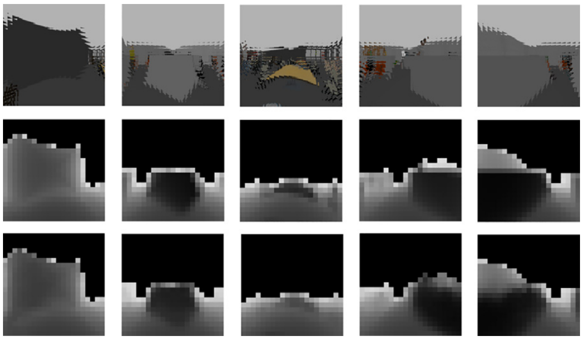

**Fig. 6.** Results of eye-wise depth estimation with compound eye images in GAZEBO dataset. From first to last row; Compound eye RGB images, ground truth compound eye depth images, estimated depth images using our method.

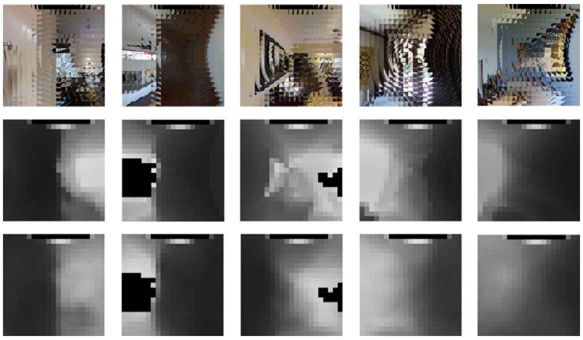

**Fig. 7.** Results of eye-wise depth estimation with compound eye images in Matterport3D dataset. From first to last row; Compound eye RGB images, ground truth compound eye depth images, estimated depth images using our method.

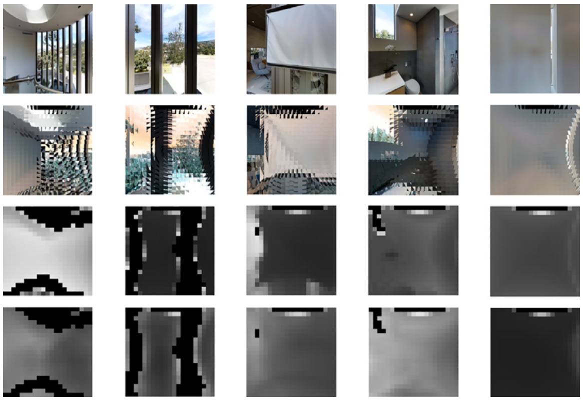

**Fig. 8.** Examples where our method fails depth estimation. From first to last row; 2D RGB images, Compound eye RGB images, ground truth compound eye depth images, estimated depth images using our method. Most failure cases are when a wall is very close or outdoors is included in the image.

**Table 3**
Comparison of 2D depth estimation methods and our method on the Matterport3D dataset.

| Method | | Accuracy | | |
|---|---|---|---|---|
| | | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ |
| **Pixel-wise** | DPT [14] | 0.7202 | 0.9306 | 0.9750 |
| | PixelFormer [27] | 0.7267 | 0.9240 | 0.9717 |
| **Patch-wise** | DPT [14] | 0.7258 | 0.9342 | 0.9779 |
| | PixelFormer [27] | 0.7432 | 0.9342 | 0.9762 |
| **Eye-wise** | DPT [14] | 0.3713 | 0.5707 | 0.7054 |
| | PixelFormer [27] | 0.3439 | 0.5374 | 0.6805 |
| | Ours | 0.5190 | 0.7885 | 0.9115 |

Our network achieves good performance in both datasets compared to other baselines in all metrics. Since all networks are trained without pre-training, most transformer-based methods show lower performance than the ResNet-based network. On the other hand, our network shows better performance than the ResNet-based network even though it uses transformer structure.

### 5.3.2. Comparison to 2D depth estimation methods

We compare our method with depth estimation state-of-the-art (SOTA) transformer-based methods learned from 2D images such as DPT [14] and PixelFormer [27] on the Matterport3D dataset. For 2D image input, we measure the performance of pixel-wise depth estimation, which predicts one depth value per pixel, and patch-wise depth estimation, which predicts one depth value per image patch of vision transformer. Additionally, for compound eye image input, we evaluate the performance of eye-wise depth estimation, which predicts one depth value per single eye. The results of comparing these three depth estimation performances with our method's eye-wise depth estimation performance can be found in Table 3.

DPT and PixelFormer show relatively high performance in pixel-wise and patch-wise depth estimation than eye-wise depth estimation. Because of the structural differences between the compound eye image and the 2D image, it can be seen that there is a significant performance difference between patch-wise depth estimation and eye-wise depth estimation, even though both are coarse depth estimations. In addition, the performance of our method is higher than that of DPT and PixelFormer in eye-wise depth estimation. This result shows that a network with a structure suitable for the compound eye is needed, like our method.

### 5.3.3. Comparison to fine-tuned ViT

Pre-training is a widely adopted technique for enhancing performance in computer vision tasks. In Section 5.3.2, DPT, which uses ViT pre-trained on the large-scale 2D image dataset and fine-tuned on the 2D image dataset, showed lower performance than ours in eye-wise depth estimation. In order to investigate the impact of pre-training on 2D images on the performance improvement in compound eye images, we combine a ImageNet pre-trained ViT model [28] with our depth head. We then compare the results of fine-tuning the combined network on compound eye images with the results of our approach. Table 4 shows the results of the comparison in the Matterport3D dataset.

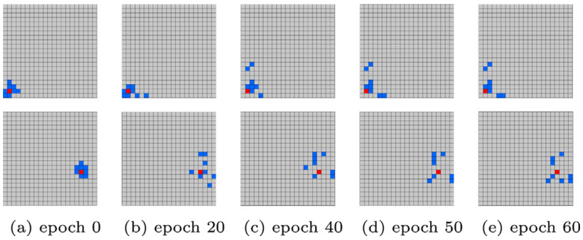

**Fig. 9.** An illustration of selected eyes in the eye selection module while learning. From left to right, it shows the selected eyes at epochs 0, 20, 40, 50, 60 in vectorized form. The two rows are the results of the different eyes of the first transformer block.

The fine-tuned ViT shows higher performance than ViT learned from scratch but lower performance than our network or ResNet-based networks learned without pre-training in Table 2. From this result, it can be seen that pre-training in 2D images does not have a significant effect on performance improvement in compound eye images, unlike in 2D images, due to differences in image structure.

### 5.3.4. Analysis of our proposed modules

We perform an ablation study to evaluate the performance of our proposed local selective self-attention and eye selection module. Table 5 shows the results of the ablation study in the GAZEBO dataset. As shown in the results, combining both modules produces the best performance.

Fig. 9 shows the convergence of the selected eyes by the proposed eye selection module during training time. It can be seen that the selected eyes for different eyes converge differently. Also in Table 5, it can be seen that the performance increases when the eye selection module is used. These results show that the eye selection module learns effective attention for each eye more than just selecting neighboring eyes which is similar to the CNN-based models.

Additionally, we compare the depth estimation performance by changing $k$ without the eye selection module to check how the performance is affected by the size of the local region. Tables 6 and 7 show the depth estimation results for different values of $k$ in the GAZEBO dataset and the Matterport3D dataset, respectively. In the GAZEBO dataset, $k = 9$ shows higher performance than others, and in the Matterport3D dataset, $k = 49$ shows higher performance. As shown in Figs. 6 and 7, the Matterport3D dataset has more complex and diverse images than the GAZEBO dataset. So, it seems that depth estimation using a wider area improves performance in the Matterport3D dataset.

### 5.3.5. Efficiency of eye-wise depth estimation

Eye-wise depth estimation is a coarse prediction that predicts the depth values of several pixels as a single value. To check the effect of eye-wise depth estimation, we compare the network that learned pixel-wise depth estimation from 2D RGB images with the network that learned eye-wise depth estimation. Using the network from [7], it is trained on the same amount of 2D images on the GAZEBO dataset. We measured the performance by averaging the values of 10 × 10 patches of the ground truth depth map and the estimated depth map. The comparison results can be seen

**Table 4**
Comparison of fine-tuned ViT and our mehtod on the Matterport3D dataset.

| Method | Accuracy | | | Error | | | | Params | FLOPs |
|---|---|---|---|---|---|---|---|---|---|
| | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ | log-mae | mae | rel | rms | (M) | (G) |
| ViT-Base (fine-tuned) | 0.4266 | 0.7337 | 0.8924 | 0.6085 | 0.3273 | 0.3475 | 0.8011 | 87.3730 | 38.3720 |
| **Ours** | **0.5190** | **0.7885** | **0.9115** | **0.5214** | **0.2867** | **0.3094** | **0.7201** | **11.2270** | **0.7687** |

**Table 5**
Ablation study on the GAZEBO dataset

| Method | Accuracy | | | Error | | | | Params | FLOPs |
|---|---|---|---|---|---|---|---|---|---|
| | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ | log-mae | mae | rel | rms | (M) | (G) |
| **Ours** | **0.7536** | **0.8606** | **0.9168** | **0.3068** | **0.2026** | **0.2875** | **0.5635** | **11.2270** | **0.7687** |
| **w/o eye selection** | 0.7473 | 0.8481 | 0.8944 | 0.3453 | 0.2275 | 0.4205 | 0.6135 | 11.2270 | 0.7687 |
| **w/o local selective self-attention** | 0.7489 | 0.8518 | 0.9063 | 0.3323 | 0.2162 | 0.3747 | 0.5997 | 11.2270 | 0.8890 |

**Table 6**
Comparison according to *k* on the GAZEBO dataset.

| | Accuracy | | | Error | | | | FLOPs |
|---|---|---|---|---|---|---|---|---|
| | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ | log-mae | mae | rel | rms | (G) |
| k=1 | **0.7476** | 0.8458 | 0.8976 | 0.3542 | 0.2320 | 0.4461 | 0.6322 | **0.7648** |
| k=9 | 0.7473 | **0.8481** | 0.8944 | **0.3453** | **0.2275** | **0.4205** | **0.6135** | 0.7687 |
| k=25 | 0.7473 | 0.8462 | **0.8977** | 0.3560 | 0.2323 | 0.4443 | 0.6337 | 0.7764 |
| k=49 | 0.7472 | 0.8455 | 0.8965 | 0.3558 | 0.2332 | 0.4485 | 0.6343 | 0.7781 |

**Table 7**
Comparison according to *k* on the Matterport3D dataset.

| | Accuracy | | | Error | | | | FLOPs |
|---|---|---|---|---|---|---|---|---|
| | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ | log-mae | mae | rel | rms | (G) |
| k=9 | 0.5130 | 0.7805 | 0.9054 | 0.5333 | 0.2929 | 0.3172 | 0.7367 | **0.7687** |
| k=25 | 0.5156 | 0.7870 | 0.9113 | 0.5255 | 0.2881 | **0.3081** | 0.7256 | 0.7764 |
| k=49 | **0.5159** | **0.7883** | **0.9118** | **0.5247** | **0.2874** | 0.3083 | **0.7229** | 0.7871 |

**Table 8**
Comparison of the performance of depth estimation and eye-wise depth estimation on the GAZEBO dataset. The performance of all three trained networks is measured when one depth value is estimated for a 10 × 10 image (image patch or single eye image).

| Method | Accuracy | | | Params | FLOPs |
|---|---|---|---|---|---|
| | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ | (M) | (G) |
| ResNet18 (Pixel-wise) | 0.4824 | 0.6554 | 0.7603 | 13.2350 | 22.7950 |
| ResNet18 (Eye-wise) | 0.7257 | 0.8401 | 0.9016 | 13.2540 | **0.2753** |
| Ours (Eye-wise) | **0.7536** | **0.8606** | **0.9168** | **11.2270** | 0.7687 |

in Table 8. The pixel-wise depth estimation network has a considerable computational cost compared to eye-wise depth estimation networks and has low performance. This result shows that eye-wise depth estimation is an efficient coarse estimation method with a small amount of computation.

## 6. Conclusion

In this work, we propose an eye-wise depth estimation network for compound eye cameras. Our network is based on the transformer architecture with local selective self-attention and eye selection modules proposed to suit the compound eye structure. With low resources, our network performs better than the ResNet-based eye-wise depth estimation network without pre-training and shows higher performance than other transformer-based structures. In addition, we find that pre-training in 2D images does not significantly affect performance improvement in compound eye images due to the distinct structure of compound eye images compared to 2D images.

To the best of our knowledge, the proposed method is the first work estimating depth by applying a transformer structure to a compound eye image. The proposed method can be applied to various vision tasks requiring depth information using compound eyes. Additionally, our approach is expected to be available as a method of recognition for various robots, such as small mobile robots, when combined with the compound eye camera hardware.

## Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Songhwai Oh reports financial support was provided by Defense Acquisition Program Administration. Songhwai Oh reports financial support was provided by Agency for Defense Development. Songhwai Oh reports financial support was provided by Korea Ministry of Science and ICT.

## Data availability

Data will be made available on request.

## Acknowledgement

This research was supported by a grant to Bio-Mimetic Robot Research Center Funded by Defense Acquisition Program Administration, and by Agency for Defense Development (UD190018ID, 50%) and the Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (No. 2019-0-01309, Development of AI Technology for Guidance of a Mobile Robot to its Goal with Uncertain Maps in Indoor/Outdoor Environments, 50%)

## References

[1] Y.M. Song, Y. Xie, V. Malyarchuk, J. Xiao, I. Jung, K.-J. Choi, Z. Liu, H. Park, C. Lu, R.-H. Kim, et al., Digital cameras with designs inspired by the arthropod eye, Nature 497 (7447) (2013) 95–99.

[2] H.L. Phan, J. Yi, J. Bae, H. Ko, S. Lee, D. Cho, J.-M. Seo, K.-i. Koo, Artificial compound eye systems and their application: A review, Micromachines 12 (7) (2021) 847.

[3] O. Cogal, Y. Leblebici, An insect eye inspired miniaturized multi-camera system for endoscopic imaging, IEEE Transactions on Biomedical Circuits and Systems 11 (1) (2016) 212–224.

[4] H. Yoo, D. Lee, G. Cha, S. Oh, Estimating objectness using a compound eye camera, in: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Nov., 2017.

[5] G. Cha, H. Yoo, D. Lee, S. Oh, Light-weight semantic segmentation for compound images, in: IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Nov., 2017.

[6] H. Yoo, G. Cha, S. Oh, Deep ego-motion classifiers for compound eye cameras, Sensors 19 (23) (2019) 5275.

[7] W. Oh, H. Yoo, T. Ha, S. Oh, Vision-based 3d reconstruction using a compound eye camera, in: 21st International Conference on Control, Automation and Systems, Oct., 2021.

[8] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R.A. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A.J. Davison, A.W. Fitzgibbon, Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera, in: Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Oct., 2011.

[9] R.A. Newcombe, D. Fox, S.M. Seitz, Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time, in: IEEE Conference on Computer Vision and Pattern Recognition, Jun., 2015.

[10] J. Lin, H. Zhu, J. Alonso-Mora, Robust vision-based obstacle avoidance for micro aerial vehicles in dynamic environments, in: IEEE International Conference on Robotics and Automation, May, 2020.

[11] D. Eigen, C. Puhrsch, R. Fergus, Depth map prediction from a single image using a multi-scale deep network, in: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems, Dec., 2014.

[12] J. Hu, M. Ozay, Y. Zhang, T. Okatani, Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries, in: IEEE Winter Conference on Applications of Computer Vision, Jan., 2019.

[13] G. Yang, H. Tang, M. Ding, N. Sebe, E. Ricci, Transformer-based attention networks for continuous pixel-wise prediction, in: IEEE/CVF International Conference on Computer Vision, Oct., 2021.

[14] R. Ranftl, A. Bochkovskiy, V. Koltun, Vision transformers for dense prediction, in: IEEE/CVF International Conference on Computer Vision, Oct., 2021.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, Dec., 2017.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: 9th International Conference on Learning Representations, May, 2021.

[17] W. Wang, E. Xie, X. Li, D. Fan, K. Song, D. Liang, T. Lu, P. Luo, L. Shao, Pyramid vision transformer: A versatile backbone for dense prediction without convolutions, in: IEEE/CVF International Conference on Computer Vision, Oct., 2021.

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: Hierarchical vision transformer using shifted windows, in: IEEE/CVF International Conference on Computer Vision, Oct., 2021.

[19] Y. Yuan, R. Fu, L. Huang, W. Lin, C. Zhang, X. Chen, J. Wang, Hrformer: High-resolution transformer for dense prediction, CoRR abs/2110.09408 (2021). https://arxiv.org/abs/2110.09408

[20] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, L. Zhang, Cvt: Introducing convolutions to vision transformers, in: IEEE/CVF International Conference on Computer Vision, Oct., 2021.

[21] Y. Li, K. Zhang, J. Cao, R. Timofte, L.V. Gool, Localvit: Bringing locality to vision transformers, CoRR abs/2104.05707 (2021). https://arxiv.org/abs/2104.05707

[22] P. Wang, X. Wang, F. Wang, M. Lin, S. Chang, W. Xie, H. Li, R. Jin, KVT: k-nn attention for boosting vision transformers, CoRR abs/2106.00515 (2021). https://arxiv.org/abs/2106.00515

[23] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, Jun., 2016.

[24] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research 15 (1) (2014) 1929–1958.

[25] N.P. Koenig, A. Howard, Design and use paradigms for gazebo, an open-source multi-robot simulator, in: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Sep., 2004.

[26] A.X. Chang, A. Dai, T.A. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, Y. Zhang, Matterport3d: Learning from RGB-D data in indoor environments, in: International Conference on 3D Vision, Oct., 2017.

[27] A. Agarwal, C. Arora, Attention attention everywhere: Monocular depth prediction with skip attention, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Jan., 2023.

[28] R. Wightman, Pytorch image models, 2019, (https://github.com/rwightman/pytorch-image-models).