Article

Sensing Range Extension for Short-Baseline Stereo Camera Using Monocular Depth Estimation

Beom-Su Seo 1, Byungjae Park 2,* and Hoon Choi 3,*

1 Intelligent Robotics Research Division, AI Research Laboratory, Electronics and Telecommunication Research Institute (ETRI), Daejeon 34129, Korea; bsseo@etri.re.kr
2 School of Mechanical Engineering, Korea University of Technology and Education, Cheonan 31253, Korea
3 Department of Artificial Intelligence, Chungnam National University, Daejeon 34134, Korea
* Correspondence: bjp@koreatech.ac.kr (B.P.); hc@cnu.ac.kr (H.C.)

Abstract: This paper proposes a method to extend a sensing range of a short-baseline stereo camera (SBSC). The proposed method combines a stereo depth and a monocular depth estimated by a convolutional neural network-based monocular depth estimation (MDE). To combine a stereo depth and a monocular depth, the proposed method estimates a scale factor of a monocular depth using stereo depth–mono depth pairs and then combines the two depths. Another advantage of the proposed method is that the trained MDE model may be utilized for different environments without retraining. The performance of the proposed method is verified qualitatively and quantitatively using the directly collected and open datasets.

Keywords: stereo camera; depth estimation; convolution neural network

1. Introduction

Range sensors are required to recognize the surrounding environment for autonomous platforms, such as autonomous vehicles, service robots, and drones. LiDAR and stereo cameras are typical range sensors that provide distance measurements, among sensors that can be mounted on platforms. LiDAR is applied on various platforms because it has a long detection range without being affected by lighting changes. However, there are two constraints to using LiDAR on a platform. The first constraint is the size of the LiDAR sensor. When the size of the platform is small, it may be difficult to mount a LiDAR sensor because of its size. The second constraint is the LiDAR sensor needs a lot of power from the platform because it must generate signals actively. In the case of large platforms, such as autonomous vehicles, there is plenty of room to install a LiDAR, and there is no problem in supplying power to the LiDAR. However, small platforms, such as service robots and drones for small delivery, may not have enough room to install LiDAR. Furthermore, the size of the battery loaded on a small platform is small, and if a LiDAR is mounted, it will cause problems such as a shorter operation time and a narrower operation range.

Unlike LiDAR, it is easy to mount a stereo camera on a small platform because it can be made passively in a small size and measure distance. Moreover, RGB images provided by the stereo camera can be used for other purposes as well, such as object detection or tracking. However, stereo cameras have two drawbacks, different from LiDAR. First, although various methods have been proposed to overcome this by adaptively controlling exposure [1], they are affected by lighting changes. Second, the distance that can be measured by stereo cameras is shorter than that of LiDAR. The method for measuring the distance from a stereo camera is to calculate the disparity using stereo matching and use the characteristics of the stereo camera, such as focal length and baseline. The baseline is a value of the stereo camera that has the greatest influence on distance measurement. The longer the baseline is, the longer the distance that the stereo camera can measure. However, if the baseline is long, the advantage of the stereo camera over LiDAR—it is small and can
be mounted on a small platform—disappears. Furthermore, even if the baseline is long, it will not detect the distance as accurately as LiDAR.

The disparity can be calculated more precisely as a way of improving the detection range without having a large baseline of the stereo camera. Recently, a convolutional neural network (CNN)-based stereo matching method such as PSMNet [2] was proposed. This method has two limitations although it can calculate the disparity precisely. First, a CNN-based stereo matching method requires many datasets to train the model and may not work well in environments different from the environment where the datasets used in the training were collected. Second, the platform must provide large memory capacity and computational power to use the CNN-based stereo matching method. Small platforms are limited in providing large memory capacity and computational power for precise disparity calculations.

In this paper, we use another approach to solve the short measurement range problem, which is a disadvantage of short-baseline stereo cameras (SBSC) used on small platforms for autonomous navigation. The proposed method combines the stereo depth (SD) of a limited range obtained with stereo matching from the SBSC and the CNN-based monocular depth estimation (MDE) [3] result. The CNN-based MDE can estimate the depth, even for a long-distance, with a monocular image only, but it has an issue that needs to be addressed when applying it to autonomous navigation. The MDE can only estimate the relative depth. A scale factor is additionally required to find the absolute depth value. A method of using the distance between the mounted position of the camera on the platform and the floor has been proposed to calculate the scale factor of MD, but there is the limitation that it cannot be used in unstructured environments [4].

To expand the SBSC's sensing range, this paper proposes a method that uses the SD of a limited range obtained from SBSC and the MD obtained from MDE to calculate the scale factor of MD and then combines the SD and MD to make the proposed depth (fused Depth:FD).

It is possible to estimate the depth with SBSC over a long distance if the recommended method is employed. The mapping, localization, and motion planning capabilities of the small platform can be improved if the proposed method is applied to an autonomous navigation system. In the case of motion planning, for example, the proposed method facilitates the detection of obstacles over longer distances, enabling safer and more efficient movements. The performance of the proposed method was validated qualitatively and quantitatively using various open and directly collected datasets.

This paper is organized as follows. Section 2 introduces various CNN-based MDE methods and compares their performances. Section 3 explains the proposed method's motivation and pipeline in detail. Section 4 introduces the experimental results obtained using the open and directly collected datasets, and Section 5 provides the conclusions and future works.

2. Convolutional Neural Network-Based Monocular Image Estimation

Structure-from-motion (SFM) is generally used to obtain the environment’s three-dimensional (3D) information using a monocular camera. SFM uses multiple images obtained from different positions as inputs. MDE, unlike SFM, can obtain the environment’s 3D information from only a single RGB image. In CNN-based MDE, a single image is used as an input, and a depth image that has the same size as the input image is output. Because CNN-based MDE can be applied to fields such as autonomous navigation, as well as augmented reality, various methods that can operate in real-time in edge devices have been actively proposed [4-7]. The aforementioned methods typically use a supervised approach with LiDAR-acquired ground truth. Some of them employ a self-supervised technique in the absence of ground truth [8]. Recently, some approaches use the transformer [9-11].

However, there is an issue when the MD obtained from CNN-based MDE is used in mapping, localization, and motion planning for autonomous navigation. When the camera parameters and environment are similar to those of the dataset used to train the
CNN model for MDE, the CNN model can estimate the depth accurately. In the meantime, if the environment and camera are not the same as in the training dataset, the absolute distance in the depth estimation result is inaccurate, though the relative distance is accurate. Using AdaBins [7], a method that shows the best performance among the recently proposed CNN-based MDE methods, we confirmed that this issue exists. When an AdaBins model trained using the KITTI training dataset [12] is applied to the KITTI test dataset, it is found that the AdaBins model can estimate the depth accurately, even for a long distance (40 m). Figure 1 shows the depth estimation results of the KITTI test dataset for the AdaBins model trained with the KITTI training dataset. In Figure 1, the red regions indicate the point groups corresponding to certain depth ranges (5–10 m, 10–20 m, and 20–40 m). As shown in the figure, the AdaBins model estimates depths up to about 40 m.

![Figure 1](image1.png)  
**Figure 1.** Depth estimation results of the KITTI test dataset for the AdaBins model trained using the KITTI training dataset (left: ground truth, right: depth estimation result). The regions shown in red in the images indicate the point groups corresponding to certain depth ranges. **(a,b)** 5–10 m, **(c,d)** 10–20 m, **(e,f)** 20–40 m.

However, the depth estimation results are inaccurate when the model trained with the KITTI training dataset is applied to a different environment or a different camera. Figure 2 shows the results of applying two well-known models (BTS [5], AdaBins [7]) trained with the KITTI training dataset to the depth estimation of datasets (CNU(Chungnam National University), UASOL [13], RADIATE [14]) collected using different cameras in different environments. The CNU dataset was collected using Intel RealSense D435 (baseline: 50 mm) on a university campus in South Korea. The UASOL dataset [13] was collected using ZED (baseline: 120 mm) on a university campus in Spain. For the RADIATE dataset [14], the same camera as the UASOL dataset was used, but the images were captured in urban and suburban areas in the United Kingdom under various lighting and weather conditions. As shown in Figure 2, the depth estimation results of the models trained using the KITTI training dataset are quite different from the actual results. In Figure 2, the red dots and regions indicate a point group existing with the depth estimation range of the stereo camera used to collect the dataset: a 10 m region for the CNU dataset and a 20 m region for the UASOL and RADIATE datasets, respectively.

To expand the sensing range of the SBSC using the off-the-shelf approach without a process of retraining the MDE model again, the scale factor of MD must be estimated first. In the next section, we describe the method for estimating the MD’s scale factor without additional training of the model and explain the method of combining the SD obtained using this and the MD of a limited range obtained from the SBSC.
Figure 2. Results of using the models (BTS, AdaBins) trained with the KITTI training dataset for depth estimation of other datasets (from left to right: ground truth, BTS depth result, AdaBins estimation result). The red dots and regions in the ground truth indicates a point group existing in the depth estimation range of the stereo camera used to collect the dataset. The red regions in the BTS and AdaBins depth estimation results indicate the point groups corresponding to depth ranges. (a–c) CNU dataset (5 to 10 m), (d–f) UASOL dataset (upper: 5 to 10 m; lower: 10 to 20 m), (g–i) RADIATE dataset. (upper: 5 to 10 m; lower: 10 to 20 m).

3. Sensing Range Extension

3.1. Overview

The scale factor of MD must be estimated first to expand the recognition range of SBSC without additional training using the MDE model. Assuming that the scale factor can be represented in some form of function regardless of environment or camera, it is possible to map the relative depth value inferred from the MD to the absolute value. Furthermore, if an appropriate parameter of an appropriate scale factor function can be found using the proposed method, the MDE model can be applied to a different environment and camera without new training whenever the environment or camera changes.

To confirm that the scale factor can be represented in some form of function and only the parameter varies depending on the environment and camera, we analyzed the results using the MDE models trained with the KITTI training dataset to estimate the depth of other datasets. Figure 3 shows graphs comparing the depth of other datasets estimated by the model trained using the KITTI training dataset and the depth obtained using LiDAR or stereo matching. In the figure, the x-axis represents the actual distance that calculated the depth using LiDAR or stereo matching. The y-axis represents the mean of MD corresponding to the actual distance. In the case of KITTI and RADIATE datasets, the actual distance was calculated using LiDAR, and in the case of CNU and UNASO datasets, the actual distance was obtained using stereo matching. The actual distance measurement range using stereo matching is shorter than that of LiDAR because of the stereo camera’s characteristics. The estimated MD of the KITTI test dataset is found to be almost the same as the actual depth because the environment and camera learned are identical. However, the estimated MD of other datasets is very different from the actual depth. Moreover, the MD obtained through the MDE model and the actual depth have strong linearity. This linearity
is common in BTS and AdaBins models. Therefore, the scale factor can be represented as a linear function, and it is possible to estimate the linear function’s parameter using the SD and MD estimated by the MDE mode.

![Figure 3. Mono depth estimation results for KITTI, CNU, RADIATE, and UASOL datasets using three models. Each model trained using KITTI datasets. (a) BTS, and (b) AdaBins.](image)

The process of expanding the recognition range of SBSC using the CNN-based MDE model proposed in this paper as shown in Figure 4 is as follows.

1. The SD depth ($X_s$) is estimated from the stereo camera.
2. MD ($X_m$) is estimated using the MDE model.
3. SD and MD pairs ($X_{inner_m}, X_{inner_s}$) are sampled in a reliable depth range.
4. The scale factor’s parameter ($M$) is estimated.
5. The scale of the MD outside of the reliable depth ($X_{outer_m}$) is estimated using the scale factor.
6. The SD depth and the scale factor-considered MD are combined to create the expanded depth (FD).

Along with the proposed method, we also used a method removing incorrect depth values existing in the sky region. Because of the lack of texture, ghosts appear in the sky region. These ghosts appear as unknown obstacles, which may cause problems with accurate environment recognition. To remove them, a CNN model for semantic segmentation is used. The depth value of a pixel that belongs to the sky region is not used to create MD–SD pairs.

![Figure 4. Pipeline of the sensing range extension method.](image)

3.2. Stereo-Mono Depth Pairing

As shown in Figure 3, the relationship between the actual depth and the MD estimated using the MDE model can be expressed as a linear function. Therefore, the SD of the limited
range obtained through stereo matching and the MD estimated using the MDE model can be used to estimate the scale factor’s parameter. From the SD and MD, reliable SD–MD pairs must be created to estimate the scale factor’s parameter. In the KITTI training data, ground truths of the close area (5 m) are almost non-existent because of the configuration of the platform used when collecting the data. Therefore, MD estimation accuracy cannot be computed closer than 5 m. Furthermore, because of the structure of SBSC, there may be errors that are above a certain level in the depth values of a range greater than a certain distance (10 m) [14–16]. In the proposed method, a reliable range for creating pairs is first determined by considering the above and the camera’s performance and noise. The depth within the SD that has a value within the range \( d_{\text{min}} = 6.5 \text{ m to } d_{\text{max}} = 8.5 \text{ m} \) is used to create SD–MD pairs.

\[
p(i, j) = (x_m(i, j), x_s(i, j)), \quad d_{\text{min}} \leq d(x_s(i, j)) \leq d_{\text{max}},
\]

In the above equation, \( x_m(i, j) \) means the depth value corresponding to the pixel \((i, j)\) in MD, and \( x_s(i, j) \) means the depth value corresponding to the pixel \((i, j)\) in SD. \( d_{\text{min}} \) and \( d_{\text{max}} \) are the minimum and maximum values of the reliable depths, respectively. \( c(i, j) \) refers to the class corresponding to the pixel in the semantic segmentation results, and \( c_s \) is the class corresponding to the sky.

Algorithm 1 shows the abstracted process of the stereo-mono depth pairing. It uses the MD, SD, and semantic segmentation result \( C \) as an input to compute the stereo-mono depth pair. A mask operation is applied to improve the computation efficiency.

**Algorithm 1: Stereo-Mono Depth Pairing**

**Input:** \( X_m, X_s, C, d_{\text{min}}, d_{\text{max}} \)

**Output:** \( P \)

\[
\begin{aligned}
\text{Idx}_{\text{mask}} &\leftarrow \{ (i, j) | d_{\text{min}} \leq X_s[i, j] \leq d_{\text{max}} \} \\
S_{\text{mask}} &\leftarrow \{ (i, j) | C \neq c_s \} \\
\text{Idx}_{\text{mask}} &\leftarrow \text{Idx}_{\text{mask}} \cap S_{\text{mask}} \\
X_s &\leftarrow X_s[\text{Idx}_{\text{mask}}] \\
X_m &\leftarrow X_m[\text{Idx}_{\text{mask}}] \\
P &\leftarrow (X_m, X_s) \\
\end{aligned}
\]

**return** \( P \)

### 3.3. Scale Factor Parameter Estimation

There may be many noises in the SD–MD pairs because of the effect of the lighting or texture distribution in the RGB image. We used RANSAC, which is robust to outliers, to estimate the scale factor’s parameter. There is one issue when estimating the scale factor’s parameter using the RANSAC: although the number of data points of the SD–MD pairs within the reliable depth range is enormous, the distribution is not even. When the scale factor’s parameter is estimated using the unevenly distributed SD–MD pairs, a biased result is obtained. To prevent this, we sorted the SD–MD pairs based on the SD and performed bucketing (Algorithm 2). When bucketing, stereo depth values within a given distance range are gathered into a single stereo depth bucket \( B_s \). The corresponding monocular depth values are also gathered into a single mono depth bucket \( B_m \). RANSAC uses the mean depth of each bucket as an input to estimate the scale factor’s parameter. \( N_{\text{step}} \) along with about 900 data points on average per bucket in the case of Figure 5.
Algorithm 2: Bucketing

**Input:** $P, d_{\text{min}}, d_{\text{max}}, N_{\text{step}}$

- $B_s = \{\}$, $B_m = \{\}$
- $\delta = (d_{\text{max}} - d_{\text{min}}) / N_{\text{step}}$

**for** $i$ from 0 to $N_{\text{step}} - 1$ **do**

- $b_s[i] = \{x_s(i, j) | d_{\text{min}} + i \cdot \delta <= d(x_s(i, j)) < d_{\text{min}} + (i + 1) \cdot \delta\}$
- $b_m[i] = \{x_m(i, j) | x_m(i, j) \in p(i, j)\}$
- $B_m \leftarrow B_m \cup b_m[i]$, $B_s \leftarrow B_s \cup b_s[i]$

**end for**

**return** $B_s, B_m$

---

**Figure 5.** Detailed pipeline of the sensing range extension method. (a) RGB, Stereo and mono depths, (b) stereo-mono depth pair sampling, (c) outer stereo prediction using scale factor parameter model $M$, and (d) depth fusion.

### 3.4. Scale Factor Parameter Update

Because the moving speed is not fast for most small platforms, the appearance of the surrounding environment does not change dramatically. It is possible to reduce the computational load used by the proposed method on small platforms if this condition is used when using the proposed method. The parameter estimated in the previous frame can be used when the appearance of the surrounding environment is similar to that of the previous frame, rather than estimating the scale factor’s parameter for every frame. For this, the following equation is used to check whether the parameter estimated in the previous frame can be used in the current frame.

$$e = \frac{\sum |M_t^{-1}(x^t_m(i,j)) - x^t_s(i,j)|}{N},$$  \hspace{1cm} (2)

The above equation obtains the error between the SD of the current frame and the scaled MD using the parameter estimated in the previous frame. $M_t^{-1}$ is the parameter estimated in the previous frame, and $N$ is the number of data points of the stereo MD–SD.
pairs in the current frame. If the value of \( e \) is not larger than the threshold value, the scale factor’s parameter is reused without estimating in the current frame. If it is larger than the threshold value, the scale factor’s parameter is estimated in the current frame.

4. Experimental Results

4.1. Datasets and Experimental Setup

The directly collected dataset and open datasets were used to verify the performance of the method proposed in this paper. The CNU dataset was collected using an Intel RealSense D435 (baseline: 50 mm) on Chungnam National University (CNU), Daejeon in South Korea (Figure 6). The UASOL dataset [13] was collected using ZED (baseline: 120 mm) on a university campus in Spain. For the RADIATE dataset [14], the same camera as the UASOL dataset was used, but the images were captured in urban and suburban areas in the United Kingdom under various lighting and weather conditions.

The CNU dataset provides distance data up to 10 m, whereas the UASOL dataset does within 20 m. The CNU and UASOL data do not give a LiDAR depth to provide an accurate physical distance. In contrast, the RADIATE [14] dataset provides LiDAR depths in adverse weather conditions such as rain, fog, and snow with left and right color images but not SD. Furthermore, as the quantitative evaluation of the data collected using a short baseline camera equipped with a precise LiDAR sensor is in progress as a future study, we used the stereo depths of UASOL and RADIATE LiDAR’s within 20 m to check the performance of the proposed method. Furthermore, we introduce the qualitative results for the UASOL and CNU data containing SDs. The proposed method was implemented with CUDA 11.02 and Python 3.8.5 on an Intel Xeon(R) CPU E5-2609@1.7 G (8 cores), 64 G memory, Nvidia RTX2080Ti, and docker 20.10.5.

![Figure 6. CNU dataset acquired using Intel RealSense D435. (a) Scooter, (b) a part of university map and path for collecting data (magenta line, S and E mean start and end point respectively).](image)

4.2. Performance Evaluation

For the indirect quantitative evaluation of the proposed method, we use the trained BTS and AdaBins models and apply them to UASOL and RADIATE datasets without any modification. As a result, we verify that the pretrained model can be used as it is without collecting new data and training a model again in new environments. To do this, we compare the differences between MD and GT, and FD and GT, respectively, according to the metrics in Table 1.

We divide the distances into three categories to evaluate the proposed method’s performance using the UASOL and RADIATE datasets: <10 m, 10–15 m, and 15–20 m. In each area, the depth estimation results are shown in Table 2. To estimate MD, we employed AdaBins and BTS models trained on the KITTI dataset. These results indicate that two models do not perform well in situations other than those in which the dataset was acquired.
for training. The proposed method, on the other hand, has smaller depth estimation errors
than the two models. The proposed method estimates depths more accurately than the two
models without retraining, according to depth estimation results using the CNU dataset
(Table 3).

Table 1. Evaluation metrics.

| Errors                      | Equations                                                                 |
|-----------------------------|---------------------------------------------------------------------------|
| Average relative error (AR) | \(\frac{1}{|N|} \sum_{X \in N} |\hat{X}_s - X_s|/X_s\)                                                             |
| Squared relative difference (SR) | \(\frac{1}{|N|} \sum_{X \in N} (|\hat{X}_s - X_s|)^2/X_s\)                   |
| Root mean squared error (RS) | \(\sqrt{\frac{1}{|N|} \sum_{X \in N} (|\hat{X}_s - X_s|)^2/X_s}\)         |
| Average Log 10 error (L10)  | \(\frac{1}{|N|} \sum_{X \in N} |\log_{10}\hat{X}_s - \log_{10}X_s|\)                                    |

Table 2. Comparison of performances on UASOL and RADIATE datasets within 20 m.

| DB   | Network | GT vs. | Evaluation Results |
|------|---------|--------|--------------------|
|      |         |        | AR (D10)           | SR (D10) | RS (D10) | L10 (D10) |
| UASOL| BTS     | MD     | 0.35               | 4.54     | 6.64     | 0.17       |
|      |         | FD     | 0.14               | 2.31     | 4.67     | 0.05       |
|      | AdaBins | MD     | 0.30               | 3.02     | 5.74     | 0.12       |
|      |         | FD     | 0.13               | 1.82     | 4.40     | 0.04       |
| RADIATE| BTS    | MD     | 0.56               | 5.50     | 8.25     | 0.19       |
|      |         | FD     | 0.09               | 0.89     | 5.19     | 0.03       |
|      | AdaBins | MD     | 0.44               | 4.06     | 7.04     | 0.16       |
|      |         | FD     | 0.1                | 0.21     | 4.56     | 0.022      |

1 D10: under 10 m, 2 D15: [10–15] m, 3 D20: [15–20] m.

Table 3. Comparison of performances on CNU dataset within 10 m.

| DB   | Network | ST vs. | Evaluation Metrics |
|------|---------|--------|--------------------|
|      |         |        | AR (D10) | SR (D10) | RS (D10) | L10 (D10) |
| CNU  | BTS     | MD     | 0.653     | 3.85     | 5.186    | 0.206     |
|      |         | FD     | 0.153     | 0.37     | 1.481    | 0.043     |
|      | AdaBins | MD     | 0.653     | 4.106    | 5.302    | 0.202     |
|      |         | FD     | 0.147     | 0.294    | 1.013    | 0.042     |

1 ST: RealSense camera depth. 2 D10: under 10 m.

We use the CNU and UASOL datasets to qualitatively demonstrate that the proposed
method expands the recognition range of SBSC. To exclude the parts belonging to the
sky region in the process of estimating the scale factor’s parameter, we used the DeepLab
v3 model [17] trained using the CityScapes dataset [18] with the backbone of MobileNet
v2 [19]. Figure 7 shows examples of the results of expanding the sensing range of the stereo
camera that has a short baseline using the proposed method. In the SD shown in Figure 7b,
the depth of the close region is estimated accurately, but the depth of a far region is not
estimated. As shown in Figure 7c, the proposed method can estimate the depth to a further
region than the SD. The regions marked with a black rectangle in Figure 7b–d, respectively,
are the sky. Because of the lack of texture in the SD calculation process, ghosts appear in the sky region. In the case of MD, there are more ghosts in the sky than in the case of SD. Figure 7d shows the result of the applied sky removal using semantic segmentation to the proposed method. The proposed method and semantic segmentation can be used to estimate the depth even for a long distance, and the ghosts appearing in the sky region can also be removed.

Figure 7. Sensing range extension result. (a) RGB, (b) stereo depth with sky in black box, (c) depth fusion without sky using semantic segmentation, and (d) depth fusion (Purple is near yet yellow is far distance).

The 3D visualization results of the depths estimated using the proposed method using the UASOL and CNU datasets, respectively, are shown in Figures 8 and 9. They show that the proposed method can be used to estimate the depth of a long-distance by expanding the sensing range of the SBSC. Figures 8d,f, and 9d,f are the results of visualizing the SD that the SBSC obtained as the 3D mesh and 3D voxel. They confirm the limitation that the SD obtained by the SBSC can be used to recognize the structure of the environment in a close region. Figures 8e,g and 9e,g show the results of visualizing the depths estimated using the proposed method as the 3D mesh and 3D voxel. They show that if the proposed method is used, the structure of the environment can be recognized, even for the region of a long distance. Furthermore, it is confirmed that semantic segmentation removes the ghosts in the sky region, resulting in no unknown obstacles.

Figure 8. 3D reconstruction results using UASOL dataset. (a) RGB, (b) stereo depth, (c) mono depth, (d) stereo depth mesh, (e) depth fusion mesh, (f) stereo depth voxel, and (g) depth fusion voxel.
Figure 9. 3D reconstruction results using CNU dataset. (a) RGB, (b) stereo depth, (c) mono depth, (d) stereo depth mesh, (e) depth fusion mesh, (f) stereo depth voxel, and (g) depth fusion voxel.

In Figures 10–12, the structure of the environment outside the range that the SBSC can be sensed using the proposed method. The red regions represent the point groups within certain ranges (5–10 m, 10–20 m, 20–40 m).

Figure 10. Depth estimation results using CNU dataset. (a) Stereo depth 5 to 10 m, (b) stereo depth 10 to 20 m, (c) stereo depth 20 to 40 m, (d) depth fusion (5 to 10 m), (e) depth fusion (10 to 20 m), and (f) depth fusion (20 to 40 m).
Figure 11. Depth estimation results using RADIATE dataset. (a) LiDAR depth 5 to 10 m, (b) LiDAR depth 10 to 20 m, (c) LiDAR depth 20 to 40 m, (d) depth fusion (5 to 10 m), (e) depth fusion (10 to 20 m), and (f) depth fusion (20 to 40 m).

Figure 12. Depth estimation results using UASOL dataset. (a) Stereo depth 5 to 10 m, (b) stereo depth 10 to 20 m, (c) stereo depth 20 to 40 m, (d) depth fusion (5 to 10 m), (e) depth fusion (10 to 20 m), and (f) depth fusion (20 to 40 m).

4.3. Computational Performance

Based on the strategy proposed in Section 3.4, we measured the performance of the proposed method by not having to estimate the scale factor’s parameter for every frame. Since the environment structure may vary, the frequency of reusing the scale factor parameter obtained in the previous frame may vary. Therefore, the RADIATE and KITTI datasets were also used, in addition to the UASOL and CNU datasets for the test. Table 4 shows the result for each dataset with 0.5 m as the threshold value. The frequency of parameter estimation for the scale factor varies based on the dataset, according to this statement. Because dissimilar images were used in nine places on campus with different landscapes, the UASOL dataset took the longest time to estimate the scale factor parameter, causing many structural changes in the dataset. The other three datasets, on the other hand, do not show major changes in the environment structure because they are relatively constant at a campus or city center. As a result, once the scale factor parameter has been computed, it can be used over multiple frames. The computing time was mostly used to estimate the monocular depth during the entire process of evaluating depths. If faster computing speed is required, a method, such as FastDepth [20], that has faster processing speed than BTS and AdaBins can be used.
Table 4. Computational performance.

| Network | Measure | CNU (640 × 480) | UASOL (1216 × 352) | RADIATE (640 × 480) | KITTI (1216 × 352) |
|---------|---------|------------------|--------------------|---------------------|--------------------|
| BTS     | # of SFPE 1/sample | 4/1760           | 74/205             | 7/719               | 0/433              |
|         | SFPE time (µ/σ) (s) | 0.014/0.004      | 0.061/0.055        | 0.013/0.009         | 0.020/0.004        |
|         | MDE time (µ/σ) (s)  | 0.159/0.004      | 0.432/0.100        | 0.160/0.007         | 0.215/0.007        |
|         | Total (s)         | 0.173            | 0.493              | 0.173               | 0.235              |

| AdaBins | # of SFPE/sample | 9/1760          | 66/205             | 0/719               | 0/433              |
|         | SFPE time (µ/σ) (s) | 0.014/0.005    | 0.057/0.057        | 0.012/0.003         | 0.019/0.004        |
|         | MDE time (µ/σ) (s)  | 0.201/0.002     | 0.230/0.004        | 0.203/0.002         | 0.227/0.002        |
|         | Total time (s)     | 0.215           | 0.287              | 0.215               | 0.246              |

1 SFPE: Scale Factor Parameter Estimation.

5. Conclusions and Future Works

To expand the SBSC’s sensing range, this paper proposes a method that takes the SD of a limited range obtained from the SBSC and calculates the scale factor of the MD obtained using the MDE to combine the two depth estimations in an FD. If it is used, we can possibly recognize the environment up to a long-distance using the SBSC. An additional advantage of ours is that the trained MDE model can be used without retraining for different data in different environments. Usually, if the MDE model is used without retraining in an environment different from the trained, the relative depth may differ significantly from the actual one. Usually, it brings tedious data manipulation (acquisition, annotation, preprocessing, etc.) with multi-channel LiDAR and SBSC. However, the proposed method can calibrate the MD by estimating the scale factor’s parameter.

Table 5 shows the pros and cons between the various methods and ours for obtaining distance data that can be importantly utilized by small platforms.

Table 5. Comparison of depth acquisition methods for small mobile robot.

| Depth Methods | Pros | Cons |
|---------------|------|------|
| LiDAR         | accurate | sparse, high cost and energy, relatively more mounting space |
| Stereo 1      | dense, relatively low cost and energy, less mounting space | accurate only within short distance 3 |
| Monocular 2   | same with stereo depth | require data manipulation 4 and retraining of CNNs if not like with train environments |
| Proposed Method | same with stereo depth, no manipulation, and retraining | relatively inaccurate if used in the train environments compared with MDE |

1 Short baseline camera. 2 Based on CNNs. 3 Dependent upon its specification. 4 Data manipulation: gathering, annotation, preprocessing, etc.

The method can be used to improve mapping, path planning, or obstacle avoidance performance by recognizing the environment, even for a long distance, on a small platform where a sensor with a large size and large power consumption, such as LiDAR, cannot be installed.

Unfortunately, we could not construct a dataset capable of quantitatively verifying the depth estimation accuracy of it and performed some qualitative validations with limited quantitative ones. In a future study, we will build a dataset to validate the depth estimation accuracy more quantitatively. Furthermore, we plan to apply it to path planning.
Author Contributions: Conceptualization, B.P.; methodology, B.-S.S. and B.P.; software, B.-S.S.; validation, B.-S.S., B.P. and H.C.; resources, B.-S.S. and H.C.; data curation, B.P.; writing—original draft preparation, B.P.; supervision, B.P. and H.C.; funding acquisition, B.-S.S. and B.P. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the ICT R&D program of MSIT/IITP[2020-0-00857, Development of Cloud Robot Intelligence Augmentation, Sharing and Framework Technology to Integrate and Enhance the Intelligence of Multiple Robots] and Korea University of Technology and Education: New professor research program of KOREATECH in 2020. H. Choi was supported by research fund of Chungnam National University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Kim, J.; Cho, Y.; Kim, A. Exposure Control using Bayesian Optimization based on Entropy Weighted Image Gradient. In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, Australia, 21–25 May 2018.
2. Chang, J.-R.; Chen, Y.-S. Pyramid Stereo Matching Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 January 2018; pp. 5410–5418.
3. Eigen, D.; Fuhrich, P.; Fergus, R. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. arXiv 2014, arXiv:1406.2283.
4. McCraith, R.; Neumann L.; Vedaldi, A. Calibrating Self-supervised Monocular Depth Estimation. arXiv 2020, arXiv:2009.07714.
5. Lee, J.H.; Han, M.-K.; Ko, D.W.; Suh, I.H. From big to small: Multi-Scale Local Planar Guidance for Monocular Depth Estimation. arXiv 2019, arXiv:1907.10326.
6. Song, M.; Lim, S.; Kim, W. Monocular Depth Estimation Using Laplacian Pyramid-Based Depth Residuals. IEEE Trans. Circuits Syst. Video Technol. 2021, 31, 4381–4393. [CrossRef]
7. Bhat, S.F.; Alhashim, I.; Wonka, P. AdaBins: Depth Estimation Using Adaptive Bins. arXiv 2021, arXiv:2011.14141.
8. Guizilini, V.; Ambrus, R.; Pillain, S.; Raventos, A.; Gaidon, A. 3D Packing for Self-Supervised Monocular Depth Estimation. arXiv 2019, arXiv:1905.02693.
9. Li, Z.; Wang, X.; Liu, X.; Jiang, J. BinsFormer: Revisiting Adaptive Bins for Monocular Depth Estimation. arXiv 2022, arXiv:2204.00987.
10. Li, Z.; Wang, X.; Liu, X.; Jiang, J. DepthFormer: Exploiting Long-Range Correlation and Local Information for Accurate Monocular Depth Estimation. arXiv 2022, arXiv:2203.14211.
11. Ranftl, R.; Bochkovskiy, A.; Kolton, V. Vision Transformers for Dense Prediction. arXiv 2021, arXiv:2103.13413.
12. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision Meets Robotics: The Kitti Dataset. Int. J. Rob. Res. 2013, 32, 1231–1237. [CrossRef]
13. Bauer, Z.; Gomez-Donoso, F.; Cruz, E.; Orts-Escolano, S.; Cazorla, M. UASOL, A Large-Scale High-Resolution Outdoor Stereo Dataset. Sci. Data 2019, 6, 162. [CrossRef] [PubMed]
14. Sheeny, M.; Pellegrin, E.D.; Saptarshi, M.; Ahrabian, A.; Wang, S.; Wallace, A. RADIATE: A Radar Dataset for Automotive Perception. arXiv 2020, arXiv:2010.09076.
15. Ahn, M.S.; Chae, H.; Noh, D.; Nam, H.; Hong, D. Analysis and Noise Modeling of the Intel RealSense D435 for Mobile Robots. In Proceedings of the International Conference on Ubiquitous Robots, Jeju, Korea, 24–27 June 2019.
16. Halmetschlag-Funek, G.; Suchi, M.; Kampel, M.; Vincze, M. An Empirical Evaluation of Ten Depth Cameras: Bias, Precision, Lateral Noise, Different Lighting Conditions and Materials, and Multiple Sensor Setups in Indoor Environments. IEEE Robot. Autom. Mag. 2019, 26, 67–77. [CrossRef]
17. Chen, L.-C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking Atrous Convolution for Semantic Image Segmentation. arXiv 2017, arXiv:1706.05587.
18. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2017.
19. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv 2016, arXiv:1704.04861.
20. Wofk, D.; Ma, F.; Yang, T.J.; Karaman, S.; Sze, V. Fastdepth: Fast monocular depth estimation on embedded systems. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 6101–6108.