Non-learning Stereo-aided Depth Completion under Mis-projection via Selective Stereo Matching

Yasuhiro Yao, Ryoichi Ishikawa, Shingo Ando, Kana Kurata, Naoki Ito, Jun Shimamura, and Takeshi Oishi

Abstract—We propose a non-learning depth completion method for a sparse depth map captured using a light detection and ranging (LiDAR) sensor guided by a pair of stereo images. Generally, conventional stereo-aided depth completion methods have two limitations. (i) They assume the given sparse depth map is accurately aligned to the input image, whereas the alignment is difficult to achieve in practice. (ii) They have limited accuracy in the long range because the depth is estimated by pixel disparity. To solve the abovementioned limitations, we propose selective stereo matching (SSM) that searches the most appropriate depth value for each image pixel from its neighborly projected LiDAR points based on an energy minimization framework. This depth selection approach can handle any type of mis-projection. Moreover, SSM has an advantage in terms of long-range depth accuracy because it directly uses the LiDAR measurement rather than the depth acquired from the stereo. SSM is a discrete process; thus, we apply variational smoothing with binary anisotropic diffusion tensor (B-ADT) to generate a continuous depth map while preserving depth discontinuity across object boundaries. Experimentally, compared with the previous state-of-the-art stereo-aided depth completion, the proposed method reduced the mean absolute error (MAE) of the depth estimation to 0.65 times and demonstrated approximately twice more accurate estimation in the long range. Moreover, under various LiDAR-camera calibration errors, the proposed method reduced the depth estimation MAE to 0.34-0.93 times from previous depth completion methods.

Index Terms—Computer vision, Depth completion, LiDAR, Sensor fusion, Stereo matching

I. INTRODUCTION

DEPTH measurement is conducted in several ways such as time-of-flight (ToF), stereo cameras, and structured light projection [1]. Stereo cameras and structured light projection estimate depth by pixel disparity. Hence their precision dramatically reduces as the distance increases since a small disparity change indicates a large depth change in the long range. In comparison, ToF sensors have a higher precision in the long range. Among ToF sensors, light detection and ranging (LiDAR) is used in various systems that require adaptability to dynamic environments, e.g., automated driving and robots, because of its active sensing capability and robustness to environmental changes. However, in terms of measurement density, LiDAR has a limitation because of the number of lasers in its array and the narrow beam measurement.

An alternative to address the sparsity of LiDAR is the depth completion, and the most common approach uses a single synchronized image as a guide [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22]. These methods generate a sparse depth map by projecting LiDAR points onto the image, and then the depth map is completed using pixel intensities. Such image-aided depth completion methods are extensively studied and range from non-learning to supervised methods.

An important issue in the depth completion is mis-projection where LiDAR points are projected onto different objects in the image. Mis-projection often occurs because of spatial and temporal displacement of sensors, dynamic objects, decalibr-
tion, and calibration errors. Because camera and LiDAR are typically placed at different positions, occlusion is unavoidable for near or dynamic objects. The temporal displacement of each LiDAR beam also causes errors without precise synchronization. Decalibration appears at run time because of oscillations of the vehicle or other mechanical reasons. Furthermore, the extrinsic calibration error between sensors results in mis-projection over the entire image.

Generally, extrinsic LiDAR and camera calibration is difficult because of their different modalities. To build the KITTI dataset [23], Geiger et al. calibrated LiDAR and cameras using multiple calibration boards in a controlled garage environment [24] followed by the manual selection of corresponding points. Although marker-less calibration methods have been examined [25], [26], [27], it remains difficult to stably realize accurate extrinsic calibration in uncontrolled environments.

In recent studies, stereo images have been used for depth completion to solve the mis-projection issue rather than a single image. This is because stereo camera systems are widely available and such systems can perceive 3D information with the help of stereo matching algorithms. For example, a self-supervised method is applicable under mis-projection caused by displacements of sensors [28]. However, this method still requires a dataset measured with a well-calibrated LiDAR-stereo system to train the neural network (NN). Furthermore, this method estimates depth by pixel disparity and suffers from low depth precision in the long range.

Therefore, in this study, we propose a non-learning depth completion method for a stereo-LiDAR system that is effective for the long range and robust to mis-projection. The proposed method comprises two techniques, i.e., selective stereo matching (SSM) and binary anisotropic diffusion tensor (B-ADT) [8]-aided smoothing. An important proposal is SSM, which searches for an optimal depth value for each pixel from its neighborly projected LiDAR points using an energy minimization framework (Fig. 1). This energy minimization approach can handle any type of mis-projection. Furthermore, SSM directly uses LiDAR depths and is advantageous in long-range accuracy. SSM is discrete optimization; thus, we apply B-ADT-aided smoothing for continuous depth estimation while preserving discontinuity between different objects.

Our contributions are summarized as follows.

- We propose SSM, which performs stereo matching in a selective manner to upsample LiDAR depths while maintaining the depth precision of LiDAR in the long range and considering mis-projection of LiDAR points.
- We propose a non-learning depth completion framework that combines SSM and B-ADT-aided smoothing. The framework achieves boundary-aware continuous depth estimation in addition to the SSM effects (long-range depth accuracy and robustness to mis-projection).

The rest of the paper is organized as follows. Section II reviews the related works and limitations of existing depth completion methods. Section III explains the proposed method. Section IV shows the evaluations. Section V gives the conclusion and summarizes limitations and future works.

II. RELATED WORK

A. Stereo matching

Stereo matching is extensively studied for 3D scanning because of the availability of stereo camera systems. The methods span from non-learning [29], [30], [31], [32], [33], [34] to NN-based self-supervised [35] and supervised [36] methods.

In terms of accuracy, the supervised methods perform the best among them. The supervised methods also have the potential to perform in the long range because the precision of the ground truth disparity, which is usually made by other sensors such as LiDAR, is high and sub-pixel-level accurate. However, it is challenging to prepare a large dataset with ground truth disparity to train NNs.

Non-learning and self-supervised methods have difficulties achieving high precision in the long range because their depth estimation is limited by pixel-level stereo matching. As mentioned, a small disparity change indicates a large depth change in the long range. Therefore, there is uncertainty in the depth estimation even if the matching is accurate at the pixel level. Moreover, stereo matching methods still suffer from challenging scenarios such as repetitive pattern, low texture, discontinuity to cause occlusion, and specular reflection conditions.

B. Single-image-aided depth completion

Depth completion methods generate high-resolution and dense depth maps from sparse or low-resolution depth maps captured using LiDAR or depth cameras. The most common approach uses a single image as guidance. Kopf et al. [2] proposed a method to interpolate low-resolution depth values based on the joint distance of color and space in a high-resolution image. Diebel and Thrun performed upsampling using a Markov random field (MRF) formulation [4]. In this method, the smoothness term is weighted as per texture derivatives; however, the results suffer from surface over flattening. To address this issue, Ferstl et al. formalized depth completion into ADT-aided and TGV-regularized energy minimization [3]; and their method has been successfully used to smooth and optimize depth maps in more recent methods [37], [38]. Recently, Yao et al. proposed B-ADT to achieve depth completion to preserve discontinuity between different objects [8].

In addition, NNs have been applied to depth completion tasks. The most common approach is to train networks with ground truth dense depth maps [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22]. Recently, self-supervised and semi-supervised methods have been examined because it is difficult to acquire the dense ground truth. Ma et al. [6] and Wong et al. [7] proposed methods with NNs that can be self-supervised using monocular camera frames and sparse depth maps from LiDAR with motion. Yang et al. proposed a method that can train a NN by the likelihood of the observed sparse point cloud under a hypothesized depth map [39].

A major limitation of the single-image-aided depth completion is that mis-projection of LiDAR points is not considered.
C. Stereo-aided depth completion

Stereo images have been used as guides to complete the sparse measurements of LiDAR. These methods have been developed based on stereo matching, and they perform dense stereo matching using the accurate sparse depth value by LiDAR as a clue.

Badino et al. used LiDAR measurements to reduce the search space for stereo matching and provided predefined paths for dynamic programming [40]. Maddern et al. proposed a probabilistic model to fuse LiDAR and disparities by combining prior from each sensor [41], and Park et al. used NNs to learn such a model, which takes two disparities as input, i.e., one from the interpolated LiDAR and the other from semi-global matching [42]. Choe et al. recently proposed a geometry-aware stereo-LiDAR fusion network for long-range depth estimation [43]. As the same as single-image-aided methods, these methods do not consider mis-projection in given sparse depth maps.

Several recent methods have attempted to infer dense disparity maps from inaccurately projected LiDAR points with the help of stereo images. For example, Cheng et al. proposed a self-supervised method to train a NN to remove occluded background projection of LiDAR points to infer dense disparity maps [28]; however, this method does not handle incorrect projection caused by extrinsic calibration errors between the LiDAR and camera. Park et al. proposed a supervised method to train a NN to infer dense disparity maps from LiDAR inputs with extrinsic calibration errors between LiDAR and the camera[44]; however, this method requires accurately calibrated LiDAR and cameras to acquire effective training data.

Furthermore, the previous non-learning and self-supervised methods [40], [41], [28] estimate the depth by pixel disparity; thus, their depth precision is limited in the long range.

In summary, there are two major limitations in existing stereo-aided depth completion methods.

- These methods require accurate LiDAR-camera extrinsic calibration at some point in their process, which is often difficult to realize.
- The precision in the depth estimation is dramatically reduced as the distance increases because of the nature of disparity estimation using images.

The cost of the proposed method is similar to the previous studies [29], [28], whereas the approach to minimize the cost is different. The proposed method searches the minimizer by the selection from projected LiDAR depth values. The approach can handle any type of mis-projection without requiring accurate LiDAR-camera extrinsic calibration in any part of the process. Moreover, this selective approach has an advantage in the long-range precision because it directly uses LiDAR depth values.

III. PROPOSED METHOD

As shown in Fig. 2, the proposed method applies SSM followed by B-ADT-aided smoothing[8]. SSM is a discrete optimization, and its output is discrete; thus, smoothing improves the quality of the result. B-ADT allows us to incor-
porate boundary-direction-aware discontinuity in a variational approach. Both SSM and B-ADT-aided smoothing; thus, the proposed method as a whole, preserve depth discontinuity between different objects.

We give the problem statement in Section III-A, introduce SSM in Section III-B, explain B-ADT-aided smoothing in Section III-C, and describe a practical parameter tuning approach for SSM in Section III-D.

A. Problem statement

Our problem settings assume a stereo camera and LiDAR are used to capture the scene; however, the camera and LiDAR calibration contains errors. Such conditions are possibly occur because of the difficulty associated with calibration, particularly when a calibration target is not available. Our aim is to estimate a dense depth map that is aligned with the image. Using mathematical notations, the target problem is defined as follows.

We are given a pair of stereo images \((I_1 : \Omega_1 \rightarrow \mathbb{R}, I_2 : \Omega_2 \rightarrow \mathbb{R})\) and a sparse inverse depth map captured by LiDAR \((D_S : \Omega_1 \rightarrow \mathbb{R} \cup \{\phi\})\) with \(\Omega_1 \subset \mathbb{R}^2\) and \(\Omega_2 \subset \mathbb{R}^2\) being the image domains and \(\phi\) indicating an empty depth. Here, \(D_S\) is determined by projecting the LiDAR points to the image \(I_1\); however, \(D_S\) is not accurately aligned with \(I_1\) because we expect LiDAR-camera miscalibration and occlusions. Our aim is to derive a dense inverse depth map \(D : \Omega_1 \rightarrow \mathbb{R}\) aligned with the input image \(I_1\). Throughout this paper, we normalize \(I_1\) and \(I_2\) to the range \([0, 1]\), and the unit of the depth is meter.

Note that we derive an inverse depth map \(D\) rather than directly deriving the depth map. This is performed to balance the contribution of both near and distant depths \([45],[8]\). Deriving a dense inverse depth map \(D\) is equivalent to deriving a dense depth map \(D^{-1}\) or dense disparity map \(D'\). The conversions at \(x \in \Omega_1\) are given as Eq. (1), (2) using camera focal length \(f\) and stereo baseline \(b\).

\[
D^{-1}(x) = D(x)^{-1} \quad (1)
\]

\[
D'(x) = fbd(x) \quad (2)
\]

The proposed method is applicable to motion stereo images as input as in the evaluation using the Komaba dataset (Section IV-B).

In this paper, we use \(|\cdot|\) to denote the vector norm. In particular, given a vector \(p \in \mathbb{R}^K\) with \(K\) being the arbitrary number of dimensions, the norm is given as follows:

\[
|p| = \sqrt{\sum_i^K p_i^2} \quad (3)
\]

B. Selective Stereo Matching

SSM searches the most appropriate inverse depth value for every \(x \in \Omega_1\) from its neighborly projected LiDAR points. SSM comprises the candidate assignment and candidate selection steps. In the candidate assignment step, each pixel in the image is assigned a set of LiDAR inverse depth values. In the candidate selection step, SSM selects the most appropriate value from the candidate set using an energy minimization framework. Here, the energy is defined as the sum of the stereo matching cost and the smoothness regularization term.

Implementation-wise, both the candidate assignment and the candidate selection are composed of pixel-wise calculations which are parallelized on GPU.

1) Candidate assignment:

a) Initial map of candidate sets: First, SSM constructs a candidate set \(S^x(x \in \Omega_1)\), which is a set of inverse depth values in the surrounding pixels within a pre-defined radius \(r\) from \(x\) (Eq. (4)), as shown in Fig. 3.

\[
S^x = \{d | d = D_S(y) \wedge d \neq \phi \text{ where } |y-x| < r\} \quad (4)
\]

Note that we introduce an empirical approach to set \(r\) in Section III-D. If the cardinality of \(S^x\) is less than predefined threshold \(m\), the set is assumed to be empty \((S^x = \emptyset)\) to avoid selecting a value from a small number of candidates (we used \(m = 4\) in our evaluations).

b) Candidate set interpolation: To fill the pixels with non-empty candidate sets, we interpolate the candidate sets using image-guided nearest neighbor search (IGNNNS) \([8]\). IGNNNS searches the nearest neighbor by the cumulative distance of image gradients. Here, let \(\Psi\) be the set of pixels where the candidate set is empty \((\Psi = \{x | S^x = \emptyset\})\), and let \(\bar{\Psi}\) be the complement of \(\Psi\) \((\bar{\Psi} = \Omega_1 \setminus \Psi\)). We search an image-guided nearest neighbor of every \(x \in \Psi\) from \(\bar{\Psi}\) (denoted \(z\)) and update the candidate set by \(S^x = S^x\).

We derive \(z\) as Eq. (5) by denoting \(\pi(x, x)\) being the set of pixels on the grid path from a surrounding non-empty pixel \(x \in \bar{\Psi}\) to \(x\).

\[
z = \arg\min_{x \in \Psi} \min_{\pi(x, x)} \sum_{y \in \pi(x, x)} \{\|\nabla I_1(y)\|^2 + c\}, \quad (5)
\]

where \(c\) is the constant cost of the unit path length, and we set \(c = 0.04\) in our evaluations following the parameter study in the literature \([8]\).

c) Correspondence search: Following candidate set interpolation, we identify the correspondence \(x'(d) \in \Omega_2\) of \(x \in \Omega_1\). \(x'(d)\) is the location on \(I_2\) where \(x\) on \(I_1\) is warped with inverse depth value \(d\). We calculate \(x'(d)\) for all combinations of \(x\) and \(d \in S^x\). Below we denote \(x = (x_0, x_1)^T\).

If \(I_1\) and \(I_2\) are a pair of rectified binocular stereo images, using the floor function denoted as \(\lfloor \cdot \rfloor\), \(x'(d)\) is derived with camera focal length \(f\) and baseline \(b\) as follows:

\[
x'(d) = \left(\frac{x_0 - fbd}{x_1}\right) \quad . \quad (6)
\]

For motion stereo images, \(x'(d)\) is calculated using the camera intrinsic parameter \((K \in \mathbb{R}^{3 \times 3})\), rotation \((R_C \in SO(3))\), and the translation \((t_C \in \mathbb{R}^3)\) of the camera motion as follows:

\[
x'(d) = \left(\frac{\tilde{x}'_0(d) / \tilde{x}'_2(d)}{\tilde{x}'_1(d) / \tilde{x}'_2(d)}\right),
\]

where

\[
\begin{pmatrix}
\tilde{x}'_0(d) \\
\tilde{x}'_1(d) \\
\tilde{x}'_2(d)
\end{pmatrix} = K \begin{pmatrix} R_C K^{-1} \begin{pmatrix} x_0 \\ x_1 \\ d^{-1} \end{pmatrix} + t_C \end{pmatrix}. \quad (7)
\]

If there are two or more \(d \in S^x\), we only maintain the nearest from \(x\) among those in \(S^x\). Note that this pruning process is performed to realize computational efficiency of the following optimization process.
2) Candidate selection via optimization:
   a) Stereo matching cost: The stereo cost $L^x(d)$ evaluates the consistency of the inverse depth $d$ and the pair of input images at location $x \in \Omega_1$.
   
   Similar to the literature [28], we compose the stereo cost $L^x(d)$ using the sum of the photometric loss $L^\rho_p (d)$, the census loss $L^c_G (d)$, and the image gradient loss $L^g_G (d)$ with weights $\alpha_C$ and $\alpha_G$ as follows:
   
   $$L^x(d) = L^\rho_p (d) + \alpha_C L^c_G (d) + \alpha_G L^g_G (d). \quad (8)$$
   
   Here, $L^\rho_p (d)$, $L^c_G (d)$, and $L^g_G (d)$ are calculated using the warped coordinates $\chi^t(d)$ in Eq. (6) or (7) with the predefined window $W$ as follows:
   
   $$L^\rho_p (d) = \sum_{\delta \in W} \min \| I_1 (x + \delta) - I_2 (\chi^t (d + \delta)) \|_H, l_p) \quad (9)$$
   
   $$L^c_G (d) = \min \| C_1 (x) - C_2 (\chi^t (d)) \|_F, l_c) \quad (10)$$
   
   $$L^g_G (d) = \sum_{\delta \in W} \min \| \nabla I_1 (x + \delta) - \nabla I_2 (\chi^t (d + \delta)) \|_H, l_g), \quad (11)$$
   
   where $C_1$ and $C_2$ respectively represent the census transformation of $I_1$ and $I_2$ with window $W$, $\| \|_H$ denotes the Hamming distance, and $l_p, l_c$, and $l_g$ are the maximum cost values. We set the window $W$ to an $11 \times 11$ square centered at $x$. In our evaluations, we set $\alpha_C = \alpha_G = 1$ and $l_p = l_c = l_g = 0.5$.

   b) Energy definition: SSM searches the optimal depth value from $S^x$ for every $x \in \Omega_1$, which is performed using an energy minimization. Here, the energy follows the conventional stereo estimation [29]. We construct an MRF whose nodes are $x \in \Omega_1$, and the edges $E$ comprise all the pairs of adjacent pixels. The energy $E_{SSM}$ is defined by the addition of the stereo matching cost $L^x(d)$ defined in Eq. (8) and a smoothness regularization term for the inverse depth as follows:
   
   $$E_{SSM} = \sum_{x \in \Omega_1} L^x(d) + \lambda_{SSM} \sum_{e \in E} \min \{ \| \Delta_c (d) \|, l_d \}, \quad \text{with } d \in S^x. \quad (12)$$
   
   Here, $\Delta_c$ represents taking the difference across the edge, and $\lambda_{SSM}$ is the regularization weight. We empirically set $\lambda_{SSM} = 10^2$.

   c) Optimization: SSM derives a discrete inverse dense depth map ($D_{SSM}$) by minimizing the energy ($E_{SSM}$) in Eq. (12).
   
   The minimization of $E_{SSM}$ is an optimization of MRF, which we solve by Loopy Belief Propagation (LBP) [46]. In particular, by setting $X \subset \Omega_1$ as the set of four adjacent pixels of $x \in \Omega_1$, we iteratively update the message from $x$ to one of its adjacent pixels $y \in X$ by the min-sum algorithm as shown in Eq. (13) and (14). Here, we denote the iteration index as $n$, the normalized message from $x$ to $y$ as $msg^n_{x \rightarrow y}(d)$, and the message prior to normalization as $msg^0_{x \rightarrow y}(d)$.
   
   $$msg^0_{x \rightarrow y}(d) = \min_{d \in S^x} L^x(d') + \lambda_{SSM} \min \{ \| d - d' \|, l_d \}$$
   
   $$+ \sum_{x' \in X \setminus y} msg^{n-1}_{x' \rightarrow x}(d') \quad (13)$$
   
   Denoting the message after convergence as $msg^{\infty}_{x \rightarrow y}$, the optimal inverse depth value $d^\infty_{SSM}$ at $x$ is expressed as follows:
   
   $$d^\infty_{SSM} = \arg \min_{d \in S^x} \left\{ L^x(d) + \sum_{y \in X} msg^{\infty}_{y \rightarrow x}(d) \right\}. \quad (15)$$
   
   The output inverse depth map $D_{SSM}$ is assigned based on the optimal values as Eq. (16).
   
   $$D_{SSM}(x) = d^\infty_{SSM} \quad (16)$$
   
   $D_{SSM}$ is visually shown in Fig. 2.
   
   In addition, for the ground mask creation in later process (Section III-C1), we construct a map $Y_{SSM} : \Omega_1 \rightarrow \Omega_1$ to indicate the original location of the inverse depth map. Because $D_{SSM}$ is created by the selection, we know where each value of $D_{SSM}$ initially located in $D_S$. In particular, if the value of $D_{SSM}$ at $x \in \Omega_1$ is originally at $y \in \Omega_1$ in $D_S$, we set $Y_{SSM} = y$. By using equations, this assignment is expressed as Eq. (17).
   
   $$Y_{SSM}(x) = y$$
   
   such that $D_S(y) \in S^x \land D_S(y) = D_{SSM}(x). \quad (17)$$

   C. B-ADT aided smoothing

   $D_{SSM}$ is discrete because it is generated by the selection from a finite number of candidates. Here, we apply B-ADT weighted TGV smoothing from [8] to derive smooth depth with discontinuity preservation at boundaries. Below, we explain the ground detection to create the filter for B-ADT derivation, the B-ADT derivation, and the optimization. Implementation-wise, the ground detection is the RANSAC plane segmentation, B-ADT derivation is a single-step calculation, and the optimization is iterative pixel-wise and parallelized on the GPU.

   1) Ground detection: We create a ground mask to filter out occlusion boundaries that are falsely detected on the ground.

   First, we detect the ground points for the input LiDAR depth map. We convert the LiDAR depth map to the point cloud and apply the RANSAC plane segmentation [48]. The RANSAC plane segmentation iteratively searches the coefficients of a plane having the maximum number of inlier points within the given threshold $d_{rand}$, by randomly sampling three points from the point cloud to derive the plane coefficients in every iteration. For RANSAC parameters, we set $d_{rand} = 0.2$ [m] and the number of iterations as 100.

   Then, we project the inlier points of the derived plane to the image domain $\Omega_1$ and acquire the ground mask $\Gamma_S : \Omega_1 \rightarrow \{0, 1\}$, where $\Gamma_S (x) = 1$ if $x$ is the ground.

   Finally, we create a dense ground mask $\Gamma_{SSM} : \Omega_1 \rightarrow \{0, 1\}$ which is aligned with $D_{SSM}$. $\Gamma_{SSM}$ is derived by interpolating $\Gamma_S$ based on the selection performed in SSM. In particular, using $Y_{SSM}$ in Eq. (17), $\Gamma_{SSM}$ is assigned as follows:
   
   $$\Gamma_{SSM}(x) = \Gamma_S(Y_{SSM}(x)) \quad (18)$$

   In Fig 2, $\Gamma_S$ and $\Gamma_{SSM}$ are visualized by indicating the ground pixels in black.
### Table I: Depth completion results on KITTI dataset with the accurate calibration

| Method                  | Input                   | *Supervised | Processing time [s] | Error rate [%] | MAE [m] |
|-------------------------|-------------------------|-------------|--------------------|---------------|---------|
| Hernández et al. [30]   | Stereo                  | No          | 0.003              | 6.59          | 0.977   |
| Yamaguchi et al. [31]   | Stereo                  | No          | 1.947              | 4.31          | 1.021   |
| Kopf et al. [2]         | Monocular + LiDAR       | No          | 1.650              | 9.70          | 0.819   |
| Ferstl et al. [3]       | Monocular + LiDAR       | No          | 0.064              | 7.45          | 0.578   |
| Yao et al. [8]          | Monocular + LiDAR       | No          | 0.065              | 4.47          | 0.413   |
| Maddern et al. [41]     | Stereo + LiDAR          | Yes         | 3.43               | 0.399         | 0.299   |
| Ours (SSM only)         | Stereo + LiDAR          | No          | 0.999              | 3.22          | 0.356   |
| Ours                    | Stereo + LiDAR          | Yes         | n/a                | 4.84          | n/a     |
| Cheng et al. [28]       | Stereo + LiDAR          | Yes         | 1.721              | 2.17          | 0.548   |

*Yes* if the method requires accurate LiDAR camera extrinsic calibration parameters during training.

2) **B-ADT derivation:** B-ADT is pixel-wise weighting for the variational regularization term. Here, B-ADT is derived based on the occlusion boundary conditions in \( D_{SSM} \) and the ground mask \( \Gamma_{SSM} \). Occlusion boundaries are boundaries where objects are not in contact; thus, the depth values at the occlusion boundaries immediately change.

The B-ADT for each pixel is assigned based on the following two conditions: \( A \), i.e., the pixel is on a vertical occlusion boundary, and \( B \), i.e., the pixel is on a horizontal occlusion boundary. Here, a vertical occlusion boundary is a vertical line segment across which the depth is horizontally discontinuous, and a horizontal occlusion boundary is a horizontal line segment across which the depth is vertically discontinuous.

In particular, with predefined threshold \( t \), we determine a pixel \( x \in \Omega_1 \) is in \( A \) if \( |\partial_{x} D_{SSM}(x)| > t \), and in \( B \) if \( |\partial_{y} D_{SSM}(x)| > t \). To make occlusion boundaries where the adjacent depths change more than 2 m, we used \( t = 2 \) in our evaluations. Because images are defined on 2D grids, every pixel belongs to one of four sets, i.e., neither \( A \) nor \( B \), \( A \), but not \( B \) (\( A \cap B \)), \( B \), but not \( A \) (\( A \cap \bar{B} \)), and \( A \) and \( B \) (\( A \cap B \)).

Boundary detection by a single threshold can be faulty, particularly in the ground region because the ground is often a large plane parallel to the view direction with a wide depth range. Thus, we filter out occlusion boundaries that are detected on the ground by \( \Gamma_{SSM} \) in Eq. (18).

Finally, by denoting B-ADT at pixel \( x \in \Omega_1 \) as \( G^x \), we set \( G^x \) based on the boundary conditions and the ground mask as follows:

\[
G^x = \begin{cases} 
1 & \text{if } x \in \bar{A} \cap \bar{B} \text{ or } \Gamma_{SSM}(x) = 1 \\
0 & \text{if } x \in A \cap \bar{B} \text{ and } \Gamma_{SSM}(x) = 0 \\
0 & \text{if } x \in \bar{A} \cap B \text{ and } \Gamma_{SSM}(x) = 0 \\
0 & \text{if } x \in A \cap B \text{ and } \Gamma_{SSM}(x) = 0 \\
0 & \text{otherwise}
\end{cases}
\]

3) **Optimization:** We minimize the energy with B-ADT weighted TGV regularization to acquire the output of the proposed framework. By denoting the inverse depth map during optimization as \( u : \Omega_1 \rightarrow \mathbb{R} \) and the relaxation variable as \( v : \Omega_1 \rightarrow \mathbb{R}^2 \), we define the energy \( E_{TGV} \) as the sum of the data term \( C[u] \) and the smoothness term \( R[u, v] \) as follows:

\[
E_{TGV} = C[u] + R[u, v] 
\]

\[
C[u] = \int_{\Omega_1} w |u - D_{SSM}|^2 \, dx
\]

\[
R[u, v] = \int_{\Omega_1} \lambda_A |G^x (\nabla u - v)| + \lambda_B |\nabla v| \, dx,
\]

where \( w \) is the pixel-wise weight for the data term, and \( \lambda_A \) and \( \lambda_B \) are weights for the energy terms. We set \( w = D_{SSM}^{-2.5} \), \( \lambda_A = 1.0 \), and \( \lambda_B = 8.0 \) based on the literature [8].

Note that \( E_{TGV} \) is convex, and we can derive the optimums of \( u \) and \( v \) using the primal dual algorithm [49]. In the end, the output of the proposed framework (a dense inverse depth map \( D \)) is the optimum of \( u \) as Eq. (23).

\[
D = \arg \min_u E_{TGV}. 
\]

**D. Parameter setting for SSM**

SSM introduces a parameter \( r \) as the radius for candidate set search. Here, we present a practical approach to select the \( r \) value. \( r \) should be as small as possible to cover the nearest appropriate depth because the number of candidates increases as \( r \) increases, which generally leads to inappropriate selections. Furthermore, we consider two primary causes for mis-projection, i.e., LiDAR-camera calibration error and occlusion.

The mis-projection caused by calibration errors is primarily attributed to rotational errors. At the center of the image, projection error \( \sigma_{calib} \) caused by rotation error \( \theta_{calib} \) can be calculated as follows:

\[
\sigma_{calib} = f \tan \theta_{calib} \text{ [pixel]},
\]

where \( f \) is the camera focal length. Although the exact value of \( \theta_{calib} \) cannot be known, it can be practically given in several ways, e.g., the error range presented in the reference of the original calibration method or by visually observing the LiDAR points projected onto the image.

To handle mis-projection by occlusion, all pixels typically should have several candidates in the range. Empirically, we found that this can be achieved when the radius is set to cover two scanlines. The pixel distance between two scanlines \( \sigma_{scan} \) is estimated using Eq. (24) with angle \( \theta_{scan} \) between the scanlines as follows:

\[
\sigma_{scan} = f \tan \theta_{scan} \text{ [pixel]}. 
\]
We set the optimal radius $r^*$ to be maximum of $\sigma_{\text{calib}}$ and $\sigma_{\text{scan}}$ to cover mis-projection caused by calibration errors and occlusion as follows:

$$r^* = \max(\sigma_{\text{calib}}, \sigma_{\text{scan}}) \, [\text{pixel}]. \quad (26)$$

IV. EVALUATION

We performed an evaluation that used the accurate LiDAR-camera extrinsic calibration (Section IV-A), another that used erroneous LiDAR-camera extrinsic calibration (Section IV-B), and the other for the parameter study (Section IV-C). In the first evaluation, we compared the accuracy of the proposed method to that of existing state-of-the-art methods under common experimental conditions [41], [42], [28]. Moreover, we analyzed the accuracy distribution over the depth range to demonstrate the advantage of the proposed method in the long range. In the second evaluation, we examined the robustness of the proposed method against LiDAR-camera extrinsic calibration errors. In this experiment, we used the KITTI [24] and Komaba datasets [38] with added calibration errors.

In all evaluations, we implemented SSM and B-ADT aided smoothing on GPU by CUDA, and used RANSAC plane segmentation from PCL library [50] for the ground detection.

A. Evaluation with accurate calibration

We evaluated the proposed method on a subset of the KITTI dataset, which is commonly used to evaluate stereo-LiDAR fusion [28], [41], [42]. These data comprise 141 sets of left and right images, sparse LiDAR depth maps, dense disparity maps, and dense depth maps. The figure of an example frame of the KITTI dataset is in the supplementary material. Here, we used the ground truths of the dense disparity map [47] and dense depth map [51] for the evaluation. Note that the input sparse depth maps still have mis-projection caused by occlusions, although the extrinsic calibration is accurate, as shown in Fig. 7 (a). In this evaluation, we set the radius for the candidate search to $r = 5$ [pixel] for our method.
Fig. 5: Disparity estimation in challenging conditions: from top to bottom, repetitive pattern, low texture, discontinuity, and specular reflection. (a) Input left images. (b) Input LiDAR disparity maps. (c) The ground truth dense disparity maps. (d), (e), (f) left: disparity estimations, right: error maps with white indicating large error. The stereo-only method [31] has larger disparity error than LiDAR-aided methods. Moreover, our method has less error in the repetitive pattern and discontinuity conditions than the state-of-the-art stereo-aided depth completion[28].

Fig. 6: Plot of MAEs against depth ranges on KITTI with the accurate calibration. The numbers in parentheses indicate the number of occurrences. The proposed method obtained smaller MAE than Cheng’s method [28] in all the ranges. Moreover, the proposed method approximately achieved half MAE in longer ranges than 40 [m].

We compared the proposed method to non-learning stereo methods [30], [31], non-learning single-image-aided depth completion methods [3], [8], non-learning stereo-aided depth completion methods [41], and supervised stereo-aided depth completion methods [42], [28]. Note that we assume Cheng’s method [28] is a supervised because it uses an accurately calibrated dataset during training.

Implementation conditions are as follows. We used our own CUDA implementations for several methods [2], [3], [8]. We used the authors’ implementation for the non-learning stereo methods [30], [31]. We used the authors’ implementation and their trained model for Cheng’s method [28]. We referred to the results in the original papers with the same experimental conditions for methods [41], [42].

1) Overall accuracy: Table I compares the accuracy of each method, and Fig. 4 shows the visualized results. This evaluation was based on the error rate measured with the dense disparity maps and the Mean Absolute Error (MAE) measured with the dense depth maps. Here, the error rate is defined as the percentage of stereo disparity outliers that have errors greater than or equal to three pixels. The proposed method outperformed the compared methods in terms of MAE. Moreover, although the proposed method is a non-learning method, it demonstrated a competitive error rate compared to the supervised stereo-aided depth completion [42], [28].

In addition, Table I demonstrates the general advantage of LiDAR-aided methods, including our method, in relation to stereo-only methods [30], [31] in terms of depth accuracy. Furthermore, Fig. 5 shows the advantage of LiDAR-aided methods in challenging conditions as a repetitive pattern, low texture, discontinuity, and specular reflection.

2) Long-range accuracy: Figure 6 shows a breakdown of MAE against the depth range compared to Cheng’s method [28]. As shown, the difference in MAE between the proposed method and Cheng’s method [28] increased as the distance increased. Cheng’s method estimate depth by pixel disparity, and its depth precision decreases as the distance increases. In contrast, the proposed method is based on the selection of LiDAR depths and does not lose the depth precision in the long range. The accuracy in the long range is also visible by the point clouds in Fig. 4. The method of Cheng et al. [28] lost the shapes of background objects, e.g., the poles, walls, and cars, whereas the shapes of these objects were retained in the results of the proposed method.

3) Processing time: Table I also shows the processing time in our environment, which is a laptop computer running Intel Core i9 and GeForce RTX 2080. Although the processing time of our method is not in real time as methods [30], [31], [8], our method performed faster than the state-of-the-art non-learning stereo matching [31] and stereo-aided depth completion [28].

B. Evaluations with calibration errors

We evaluated the proposed method with LiDAR camera extrinsic calibration errors. Here, we applied random errors to the KITTI and Komaba datasets. This comparison was performed against unsupervised methods [2], [3], [8]. Supervised stereo-LiDAR fusion methods were not applied because...
TABLE II: Input calibration error details

| Dataset | error type  | Rot. axis (x,y,z) | Rot. error [deg.] | Trans. direction (x,y,z) | Trans. error [m] |
|---------|-------------|------------------|-------------------|------------------------|-----------------|
| KITTI   | error-1 (avg.)* | (0.04,-0.89,0.45) | 0.952             | (0.03,-0.05,-0.99)      | 0.076           |
|         | error-2 (avg.)* | (-0.79,0.44,0.43) | 0.675             | (0.51,0.54,0.67)       | 0.155           |
|         |              | (-0.94,0.33,-0.03) | 0.667             | (0.36,0.83,0.42)       | 0.207           |
| Komaba  |              | (0.51,-0.11,0.85)  | 1.096             | (-0.26,-0.73,0.63)     | 0.207           |

* Averages are shown since every frame has different errors.

TABLE III: Depth completion results obtained on KITTI dataset with calibration errors

| Method       | error-1 | error-2 |
|--------------|---------|---------|
| Ferstl et al. [3] |         |         |
| Yao et al. [8]    |         |         |
| Ours (SSM only)  |         |         |
| Ours           |         |         |

TABLE IV: Depth completion results obtained on Komaba dataset with calibration errors

| Method       | lines-16 | lines-32 | lines-64 |
|--------------|----------|----------|----------|
| Kopf et al. [2] | 6.80 × 10⁻³ | 0.924 | 6.70 × 10⁻³ | 0.921 | 6.63 × 10⁻³ | 0.910 |
| Ferstl et al. [3] | 7.76 × 10⁻³ | 1.001 | 7.68 × 10⁻³ | 1.034 | 6.64 × 10⁻³ | 1.013 |
| Yao et al. [8] | 7.16 × 10⁻³ | 0.917 | 7.24 × 10⁻³ | 0.938 | 7.23 × 10⁻³ | 0.933 |
| Ours (SSM only) | 6.52 × 10⁻³ | 0.866 | 6.17 × 10⁻³ | 0.827 | 6.05 × 10⁻³ | 0.816 |
| Ours           | 6.44 × 10⁻³ | 0.857 | 6.09 × 10⁻³ | 0.816 | 6.00 × 10⁻³ | 0.809 |

Fig. 7: Mis-projection in the KITTI dataset. (a) Mis-projection is caused by occlusion although the calibration is accurate. (b), (c), (d) Calibration errors cause mis-projection in addition to the occlusion (see the poles).

accurately calibrated scans for training are not available. In this evaluation, the parameter settings were the same as those discussed in Section IV-A, except for the candidate search radius, which was set to r = 15 [pixel].

1) KITTI dataset: We applied the following three error types to the KITTI dataset used in Section IV-A.

- **blueprint** represents the extrinsic parameters before calibration, derived by the sensor setup blueprint of the KITTI dataset.
- **error-1** represents parameters calibrated by a single-frame-marker-less method from the initial 2 [deg.] of rotation and 0.2 [m] of translation errors. After calibration, the average error was 0.675 [deg.] and 0.155 [m].
- **error-2** represents parameters calibrated by a single-frame-marker-less method from the initial 4 [deg.] of rotation and 0.4 [m] of translation errors. After calibration, the average error was 0.667 [deg.] and 0.207 [m].

The single-frame-marker-less calibration method to derive error-1 and error-2 is explained in the supplementary material. The intention of error-1 and error-2 is to emulate the worst-case calibration error expected in practical cases.

Table II shows the details of the errors, and Fig 7 shows the visual of the calibration error in the input data. Note that KITTI with calibration errors also has mis-projection caused by temporal and spatial occlusions in the original KITTI dataset.

Table III shows the results obtained on the KITTI dataset. The proposed method outperformed the baselines under all experimental conditions. Moreover, by comparing with those in Table I (the same dataset with accurate calibration), the proposed method outperformed Park’s method [42] in terms of error rate and Cheng’s method [28] in terms of MAE, although the proposed method was applied to data with calibration errors. The results indicate that the proposed method is robust to LiDAR-camera extrinsic calibration errors. Figure 8 shows the results and error maps. Figure 8 indicates that the proposed method was applied to data with calibration errors.
scans is estimated by aligning the LiDAR point clouds. There is no spatial and temporal displacement between the camera and LiDAR, and occlusions are not expected in the Komaba dataset.

To create input sparse depth maps, we sampled the original dense depth maps and applied the randomly generated calibration errors (Table II). Here, three sampling patterns were applied to simulate different LiDAR resolutions.

- **lines-16** sampled 16 scanlines.
- **lines-32** sampled 32 scanlines.
- **lines-64** sampled 64 scanlines. This condition is similar to the KITTI dataset, which has 64 scanlines.

The density and mis-projection in input data are visualized in Fig. 9.

### C. Parameter study

We evaluated the effect on MAE of the value of \( r \) using the KITTI dataset (Section IV-A), the KITTI dataset with the blueprint condition, and the Komaba dataset with the lines-64 condition (Section IV-B). The results are shown in Table V.

In Table V, the value of \( r^* \) derived from Eq. (26) reside close to the \( r \) value to give the minimum MAE for every data. Here, we derived \( r^* \) for each data using Eq. (26) as follows.

- **KITTI**: \( r^* = 6.68 \) with \( f = 959.7915 \), \( \theta_{\text{calib}} = 0 \), and \( \theta_{\text{scan}} = 0.4 \).
- **KITTI (blueprint)**: \( r^* = 15.90 \) with \( f = 959.791 \), \( \theta_{\text{calib}} = 0.952 \), and \( \theta_{\text{scan}} = 0.4 \).
- **Komaba (lines-64)**: \( r^* = 16.453 \) with \( f = 956.925 \) and \( \theta_{\text{calib}} = 1.096 \). Note that, in this case, we ignored \( \sigma_{\text{scan}} \) because occlusion was not expected.

Hence, the results supports our approach to set \( r \) in Section III-D.
Fig. 10: Inputs and results on Komaba dataset with calibration errors (top: lines-16, middle: lines-32, and bottom: lines-64). Refer Fig. 11 for the ground truth. (a) The input sparse depth maps projected onto the input image. (b), (c) Depth completion results and error maps. Around the boundary of the van in the rear left, the depth is not correctly estimated by Yao’s method [8] and the proposed method with lines-16. With more scanlines (lines-32 and lines-64), the proposed method estimated depth more accurately around the boundary.

Table V: MAE variation by changing the radius in SSM

| Radius [pixels] | KITTI | KITTI w/ error (blueprint) | Komaba w/ error (lines-64) |
|-----------------|-------|-----------------------------|-----------------------------|
| r = 2           | 0.374 | 1.575                       | 0.949                       |
| r = 5           | 0.356 | 1.157                       | 0.892                       |
| r = 10          | 0.442 | 0.565                       | 0.814                       |
| r = 15          | 0.493 | **0.528**                   | **0.809**                   |
| r = 20          | 0.530 | 0.573                       | 0.860                       |

V. Conclusion

We proposed a non-learning stereo-aided depth completion method that is robust to mis-projection and preserves LiDAR precision in the long range. Unlike previous methods, our method does not require accurate LiDAR-stereo extrinsic calibration parameters in any part of its process. Therefore, it is applicable in the conditions that the calibration is difficult to conduct. In the evaluations, our method demonstrated smaller MAEs than previous state-of-the-art stereo-aided depth completion methods.

Our proposal is composed of SSM and the framework combining SSM and B-ADT aided smoothing. SSM searches for an optimal depth value for each pixel from its neighborly projected LiDAR points by an energy minimization approach, which can handle any type of mis-projection. In addition, we apply B-ADT-aided smoothing [8] to generate boundary-preserving continuous depth maps since SSM is discrete optimization.

The current limitations of the proposed method include the accuracy dependency on the LiDAR scan density, as demonstrated by the evaluation discussed in Section IV-B.

We aim to extend our approach to run in real time for applying it to actual robotic systems. Since most of our processing time comes from LBP of SSM candidate selection (0.943 out of 0.999 [s]), we consider improving the selection process to be able to adapt faster optimizers.

References

[1] J. Batlle, E. Mouaddib, and J. Salvi, “Recent progress in coded structured light as a technique to solve the correspondence problem: a survey,” Pattern recognition, vol. 31, no. 7, pp. 963–982, 1998.
[2] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, “Joint bilateral up-sampling,” in ACM Transactions on Graphics (ToG), vol. 26, no. 3. ACM, 2007, p. 96.
[3] D. Ferschl, C. Reinbacher, R. Ranftl, M. Rüther, and H. Bischof, “Image guided depth upsampling using anisotropic total generalized variation,” in International Conference on Computer Vision (ICCV), 2013, pp. 993–1000.
[4] J. Diebel and S. Thrun, “An application of markov random fields to range sensing,” in Advances in neural information processing systems, 2006, pp. 291–298.
[5] N. Schneider, L. Schneider, P. Pinggera, U. Franke, M. Pollefeys, and C. Stiller, “Semantically guided depth upsampling,” in German conference on pattern recognition. Springer, 2016, pp. 37–48.
[6] F. Ma, G. V. Cavalheiro, and S. Karaman, “Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera,” in International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 3288–3295.
[7] A. Wong, X. Fei, S. Tsuei, and S. Soatto, “Unsupervised depth completion from visual inertial odometry,” IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 1899–1906, 2020.
[8] Y. Yao, M. Roxas, R. Ishikawa, S. Ando, J. Shimamura, and T. Oishi, “Discontinuous and smooth depth completion with binary anisotropic diffusion tensor,” IEEE Robotics and Automation Letters, vol. 5, no. 4, pp. 5128–5135, 2020.
[9] L. Liu, X. Song, X. Lyu, J. Diao, M. Wang, Y. Liu, and L. Zhang, “Fcrf-net: Feature fusion based coarse-to-fine residual learning for depth completion,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 3, 2021, pp. 2136–2144.
[10] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I.-S. Kweon, “Non-local spatial propagation network for depth completion,” in European Conference on Computer Vision, ECCV 2020. European Conference on Computer Vision, 2020.
[11] X. Cheng, P. Wang, C. Guan, and R. Yang, “Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 07, 2020, pp. 10 615–10 622.
[12] S. Zhao, M. Gong, H. Fu, and D. Tao, “Adaptive context-aware multimodal network for depth completion,” IEEE Transactions on Image Processing, 2021.
Y. Chen, B. Yang, M. Liang, and R. Urtasun, “Learning joint 2d-3d representations for depth completion,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 10023–10032.

J. Qiu, Z. Cui, Y. Zhang, X. Zhang, S. Liu, B. Zeng, and M. Pollefeys, “Deeplidar: Deep surface normal guided depth prediction for outdoor scenes from sparse lidar data and single color image,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 3313–3322.

A. Li, Z. Yuan, Y. Ling, W. Chi, C. Zhang et al., “A multi-scale guided cascade hourglass network for depth completion,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 32–40.

W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, “Sparse and noisy lidar completion with rgb guidance and uncertainty,” in 19th International Conference on Machine Vision Applications (MVA). IEEE, 2019, pp. 1–6.

Y. Xu, X. Zhu, J. Shi, G. Zhang, H. Bao, and H. Li, “Depth completion from sparse lidar data with depth-normal constraints,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 2811–2820.

L. Yan, K. Liu, and E. Belyaev, “Revisiting sparsity invariant convolution: A network for image guided depth completion,” IEEE Access, vol. 8, pp. 126323–126332, 2020.

A. Eldesokey, M. Felsberg, and S. F. Khan, “Confidence propagation through cnns for guided sparse depth regression,” IEEE transactions on pattern analysis and machine intelligence, vol. 42, no. 10, pp. 2423–2436, 2019.

R. Schuster, O. Wasenmuller, C. Unger, and D. Stricker, “Ssgp: Sparse spatial guided propagation for robust and generic interpolation,” in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2021, pp. 197–206.

L. Bai, Y. Zhao, M. Elhousni, and X. Huang, “Depthnet: Real-time lidar point cloud depth completion for autonomous vehicles,” IEEE Access, 2020.

S. S. Shivakumar, T. Nguyen, I. D. Miller, S. W. Chen, V. Kumar, and C. J. Taylor, “D fusenet: Deep fusion of rgb and sparse depth information for image guided dense depth completion,” in 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 2019, pp. 13–20.

A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” The International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, 2013.

A. Geiger, F. Moosmann, O. Car, and B. Schuster, “Automatic camera and range sensor calibration using a single shot,” in 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 3936–3943.

G. Pandey, J. McBride, S. Savarese, and R. Rustic, “Automatic target-less extrinsic calibration of a 3d lidar and camera by maximizing mutual information,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 26, no. 1, 2012.

R. Ishikawa, T. Oishi, and K. Ikeuchi, “Lidar and camera calibration using motions estimated by sensor fusion odometry,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 7342–7349.

V. John, Q. Long, Z. Liu, and S. Mitra, “Automatic calibration and registration of lidar and stereo camera without calibration objects,” in 2015 IEEE International Conference on Vehicular Electronics and Safety (ICVES). IEEE, 2015, pp. 231–237.

X. Cheng, Y. Zhong, Y. Dai, P. Ji, and H. Li, “Noise-aware unsupervised deep lidar-stereo fusion,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 6339–6348.

L. Zhang and S. M. Seitz, “Parameter estimation for mrf stereo,” in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), vol. 2. IEEE, 2005, pp. 288–295.

D. Hernandez-Juarez, A. Chacon, A. Espinosa, D. Vazquez, J. C. Moure, and A. M. Lopez, “Embedded real-time stereo estimation via semi-global matching on the gpu,” Procedia Computer Science, vol. 80, pp. 143–153, 2016.

K. Yamaguchi, D. McAllester, and R. Urtasun, “Efficient joint segmentation, occlusion labeling, stereo and flow estimation,” in European Conference on Computer Vision. Springer, 2014, pp. 756–771.

M. P. Muresan, M. Negru, and S. Nedevschi, “Improving local stereo algorithms using binary shifted windows, fusion and smoothness constraint,” in 2015 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP). IEEE, 2015, pp. 179–185.

R. Spangenberg, T. Langner, S. Adelfeld, and R. Rojas, “Large scale semi-global matching on the cpu,” in 2014 IEEE Intelligent Vehicles Symposium Proceedings. IEEE, 2014, pp. 195–201.

Yasuhiro Yao received the B.S. and the M.E. degree from the University of Tokyo, Japan, in 2007 and 2010, respectively. In 2010, he joined NTT as a researcher. From 2013 to 2016, he was a cloud solution architect at Dimension Data APAC, Singapore. He is currently a Senior Research Engineer at NTT Human Informatics Laboratories and also pursuing a Ph.D. in Information Studies at the University of Tokyo. His research interests include computer vision and sensor fusion.
Ryoichi Ishikawa received the B.E. degree from the Department of Electrical Engineering, The University of Tokyo, Japan, in 2014, and the M.E. and Ph.D. degrees in Electrical Engineering and Information Systems, The University of Tokyo, Japan, in 2016 and 2019, respectively. From 2019, he is a Project Researcher at the Institute of Industrial Science, The University of Tokyo. His research interests include robot vision, sensor fusion, and calibration.

Shingo Ando received a B.E. in electrical engineering from Keio University, Kanagawa, in 1998 and a Ph.D. in engineering from Keio University in 2003. He joined NTT in 2003. He has been engaged in research and practical application development in the fields of image processing, pattern recognition, and digital watermarks. He is a member of IEICE and the Institute of Image Information and Television Engineers.

Kana Kurata received a B.S. in earth and planetary sciences from the Nagoya University in 2016 and an M.E. in environmental studies from the Nagoya University in 2018. She joined NTT in 2018. She has been engaged in research in fields of computer vision and pattern recognition.

Naoki Ito received the M.E. from Toyohashi University of Technology, Aichi, in 2001. He joined NTT in 2001 and engaged in research on character recognition. He moved to NTT EAST in 2004 and engaged in the development of security systems. He moved to NTT Cyber Space Laboratories in 2008. He has been working on the development of real-world digitalization technologies. He is currently a senior research engineer at NTT Human Informatics Laboratories.

Jun Shimamura received a B.E. in engineering science from Osaka University in 1998 and an M.E. and Ph.D. from Nara Institute of Science and Technology in 2000 and 2006. He joined NTT Cyber Space Laboratories in 2000. He is currently senior research engineer, supervisor of scene analysis technology at NTT Human Informatics Laboratories, Japan. His research interests include computer vision and mixed reality.
Supplementary to Non-learning Stereo-aided Depth Completion under Mis-projection via Selective Stereo Matching

(a) Left image  (b) Right image
(c) Sparse depth map    (d) Dense disparity map
(e) Dense depth map

Fig. A-1: A frame of the KITTI dataset. (d) The ground truth dense disparity maps are from the stereo 2015 benchmark (e) The ground truth dense depth maps are from the depth completion benchmark

Fig. A-2: A frame of the Komaba dataset.

A-1. DATASET FIGURES

Here, we visually show the examples of the dataset: KITTI in Fig. A-1 and Komaba in Fig. A-2.

A-2. LiDAR STEREO EXTRINSIC CALIBRATION

Here, we explain the single-frame-marker-less calibration method to create error-1 and error-2 in our evaluation with calibration errors. We intended to emulate the maximum error expected in practical situations. Hence, we extrinsically calibrated LiDAR and camera by using the same inputs as our depth completion, which are a single pair of images and a single scan of LiDAR.

Specifically, we search the rotation matrix $R^*$ and translation vector $t^*$ to minimize the cost $L$ from the candidate set of rotation matrices $\mathcal{R} = \{ R_i | i = 0, 1 \ldots N_R - 1, R_i \in SO(3) \}$ and translation vectors $\mathcal{T} = \{ t_i | i = 0, 1 \ldots N_t - 1, t_i \in \mathbb{R}^3 \}$ in a brute force manner with $N_R$ and $N_t$ being the number of candidates, where $\mathcal{R}$ and $\mathcal{T}$ are generated around the initial values via grid sampling.

$$ R^*, t^* = \arg \min_{R \in \mathcal{R}, t \in \mathcal{T}} L \tag{A-1} $$

For each combination of $R \in \mathcal{R}$ and $t \in \mathcal{T}$, we first project each LiDAR point $p \in P$ to the left image by applying $R$ and $t$ to derive set of the image locations and the depth of the point $(PRt)$:

$$ PRt = \left\{ (x_0, x_1, d)^T | 0 \leq x_0 < W, 0 \leq x_1 < H, d > 0 \right\}, $$

where

$$ \begin{align*}
X_0 &= X_0/D \\
X_1 &= X_1/D \\
d &= bf/D
\end{align*} \tag{A-2} $$

Here, $W$ and $H$ are the width and height of $I_1$, respectively.

To construct the cost, we introduce the $L^x_B$ term to avoid LiDAR points to be projected on the areas where no object exists. With using the notation of the stereo matching cost $L^x(d)$ in Eq. (8) of the main paper, $L^x_B$ is defined as follows:

$$ L^x_B = -\min (L^x(0), L_B). \tag{A-3} $$

Note that $L^x_B$ increases if a LiDAR point is projected onto an image location that appears to have infinite depth (or zero disparity).

Then, the cost $L_{\text{calib}}$ for calibration is defined as the sum of $L^x(d)$ and $L^x_B$ over points in $PRt$.

$$ L_{\text{calib}} = \sum_{(x, d)^T \in PRt} L^x(d) + L^x_B \tag{A-4} $$

TABLE A-1: Errors in extrinsic parameters

| Cost       | Avg. rot. error [deg.] | Avg. trans. error [m] |
|------------|------------------------|-----------------------|
| $L^x(d)$   | 0.561                  | 1.813                 |
| $L^x(d) + L^x_B$ | 0.675                  | 0.667                 |

arXiv:2210.01436v1  [cs.CV]  4 Oct 2022
Table A-1 shows the extrinsic parameter errors after calibration with or without the $L_B^x$ term. The results indicate that the $L_B^x$ term has little effect when the initial calibration error is small as in error-1. However, the $L_B^x$ term stabilizes calibration when the initial error is large as in error-2.