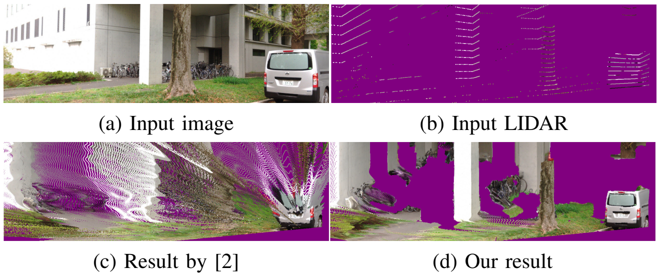
IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Discontinuous and Smooth Depth Completion with Binary Anisotropic Diffusion Tensor

Yasuhiro Yao[1,2], Menandro Roxas[1], Ryoichi Ishikawa[1], Shingo Ando[2], Jun Shimamura[2], and Takeshi Oishi[1]

***Abstract*— We propose an unsupervised real-time dense depth completion from a sparse depth map guided by a single image. Our method generates a smooth depth map while preserving discontinuity between different objects. Our key idea is a Binary Anisotropic Diffusion Tensor (B-ADT) which can completely eliminate smoothness constraint at intended positions and directions by applying it to variational regularization. We also propose an Image-guided Nearest Neighbor Search (IGNNS) to derive a piecewise constant depth map which is used for B-ADT derivation and in the data term of the variational energy. Our experiments show that our method can outperform previous unsupervised and semi-supervised depth completion methods in terms of accuracy. Moreover, since our resulting depth map preserves the discontinuity between objects, the result can be converted to a visually plausible point cloud. This is remarkable since previous methods generate unnatural surface-like artifacts between discontinuous objects.**



(a) Input image (b) Input LIDAR

(c) Result by [2] (d) Our result

Fig. 1. Depth completion results in point cloud representation. (c) Multiple objects are connected by points forming surface-like artifacts. This is because the method imposes smoothness cost across the boundaries of the object. (d) Our method successfully separates multiple objects because B-ADT enables us to eliminate smoothness cost across the boundaries.

I. Introduction

Long-range and real-time depth estimation can be achieved using Light Detection and Ranging (LIDAR) sensors. They have longer measurable range compared to depth cameras and hence commonly used for outdoor 3D scanning. But LIDAR can only give sparse measurements in real time (e.g. 10Hz Velodyne sensors [1]) due to the limited number of lasers in its array.

One way to address the sparsity of the depth data is through depth completion. It is a technique that takes a sparse depth map and other related information, such as RGB images, as inputs and infers a dense depth map.

Although there are supervised depth completion methods in the literature, we focus on unsupervised methods, among which variational methods have been successful. In [2], an image-guided Anisotropic Diffusion Tensor (ADT) is applied to the Total Generalized Variation (TGV) regularization which enabled the generation of high-resolution depth images that, although smooth, preserves object boundaries to some extent.

A disadvantage of the method in [2] is that the depth transition between objects is still continuous. The effect can be seen as points spreading between discontinuous objects in point cloud representation (Fig. 1). Such points are not plausible because they form surface-like artifacts at the location where no object exists. They also degrade the depth completion accuracy.

To overcome this, we propose a depth completion method that outputs a dense depth map that is discontinuous at occlusion boundaries and smooth elsewhere. We introduce a Binary Anisotropic Diffusion Tensor (B-ADT) and an Image-guided Nearest Neighbor Search (IGNNS). B-ADT enables us to incorporate boundary direction-aware discontinuity in variational methods. IGNNS enables us to jointly derive the occlusion boundaries, which are used for B-ADT derivation, from an image and a sparse depth map. IGNNS also serves to give a piecewise constant depth map to be used in the data term of the variational energy. We show in our experiments the advantages of depth completion with B-ADT and IGNNS against previous and baseline methods. With our implementation, we were able to achieve real-time processing using modern GPUs.

In summary, our contributions are as follows.

- We propose B-ADT applicable to variational regularization in image processing for eliminating the smoothness constraint at intended pixels and directions.
- We propose a real-time fully unsupervised depth completion method guided by a single image with B-ADT and IGNNS. The proposed method generates depth maps with apparent occlusion boundaries.
- We show in our experiments that our method can outperform previous unsupervised and semi-supervised depth completion methods in terms of accuracy. We also show that our result is suitable for point cloud representation since it preserves discontinuity between objects. Additionally, we provide the parameter sensitivity evaluation of our method.

[1] Yasuhiro Yao, Menandro Roxas, Ryoichi Ishikawa, and Takeshi Oishi are with with the Institute of Industrial Science, the University of Tokyo, Tokyo 153-8505, Japan (email: yao@cvl.iis.u-tokyo.ac.jp; roxas@cvl.iis.u-tokyo.ac.jp; ishikawa@cvl.iis.u-tokyo.ac.jp; oishi@cvl.iis.u-tokyo.ac.jp)

[2] Yasuhiro Yao, Shingo Ando, and Jun Shimamura are with the Media Intelligence Laboratories, Nippon Telegraph and Telephone Corporation, Tokyo 100-8116, Japan (e-mail: yasuhiro.yao.tc@hco.ntt.co.jp; shingo.andou.fv@hco.ntt.co.jp; jun.shimamura.ec@hco.ntt.co.jp)

II. Related Works

In this section, we discuss the scopes and limitations of existing depth completion methods including supervised and unsupervised methods.

A. Supervised Methods

Supervised methods can be classified into fully supervised [3], [4] and semi-supervised [5]–[8]. Fully supervised methods require ground truth dense depth maps for training. However, such dense depth maps are not generally available since it requires the integration of multiple sensors to produce as in [1]. On the other hand, semi-supervised methods do not require the same training overhead. Ma et al. [5] and Wong et al. [8] proposed methods with Deep Neural Networks (DNN) which can be self-supervised using the monocular camera frames and sparse depth maps from LIDAR with motion. However, the method does not apply to scenes without camera and LIDAR motion. Schneider et al. proposed a method that pre-trained semantic segmentation for guided upsampling of sparse depth maps [6], and Hirata et al. used pre-trained semantic segmentation and motion stereo for a similar purpose [7]. In principle, supervised methods apply only to scenes of the same domain as the training.

B. Unsupervised Methods

Compared with supervised methods, unsupervised methods are more applicable in situations where a large amount of data for training is not available. The earliest unsupervised depth completion methods are based on interpolation. Kopf et al. [9] proposed a method to interpolate low-resolution depth values based on the joint distance of color and space in the high-resolution image. More recently Ku et al. proposed a method by combining handcrafted classical image processing algorithms [10].

A disadvantage of the interpolation method is that it introduces undesirable smoothness to the result. To address this problem, energy minimization methods have been studied. Diebel and Thrun performed an upsampling using a Markov Random Field (MRF) formulation [11], where the smoothness term is weighted according to texture derivatives. However, the MRF enhanced method results in surface flattening. To accommodate this, Ferstl et al. formalized depth completion into ADT-aided TGV regularized energy minimization [2]. Their method was successfully used to smooth and optimize depth maps in more recent methods [7], [12].

Nevertheless, since ADT does not eliminate smoothness along occlusion boundaries, it produces surface-like artifacts between foreground and background objects. This problem becomes more obvious when we represent the depth as a 3D point cloud.

III. Our Method

Our depth completion is composed of IGNNS, boundary (B-ADT) derivation, and the energy minimization (Fig. 2). The core of our method is B-ADT. We first review ADT and its application as preliminary in Section III-A. Then, we introduce B-ADT in Section III-B, IGNNS in Section III-C, our boundary derivation in Section III-D, and our energy minimization in Section III-E.

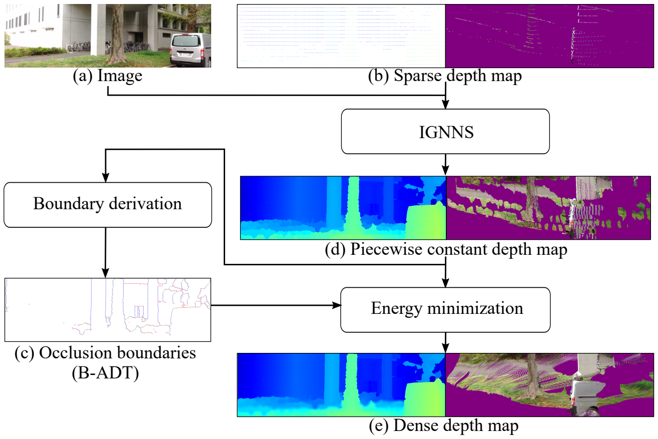


Fig. 2. Flow chart of our depth completion. (b), (d), (e) Left: depth maps, right: point clouds from a side viewpoint. (c) Colors indicate, white: no occlusion boundary, blue: vertical occlusion boundary, red: horizontal occlusion boundary, green: vertical and horizontal occlusion boundary. (d) In the piecewise constant depth map, all surfaces are parallel to the image plane. (e) Our energy minimization smooths depth within an object while preserving discontinuity at object boundaries.

We formalized the depth completion problem as follows. Given an image $I : \Omega \to \mathbb{R}$ and a sparse depth map $d : \Omega \to \mathbb{R}$ where $\Omega \subset \mathbb{R}^2$ is the image domain, our goal is to find the upsampled depth map $u : \Omega \to \mathbb{R}$. In our implementation, I is a gray-scale image normalized to the range $[0, 1]$.

A. Preliminary: Anisotropic Diffusion Tensor

ADT was proposed by Werlberger et al. in [13] for optical flow estimation and applied to other problems such as stereo matching [14] and depth completion [2]. ADT serves the purpose of an anisotropic weighting of the regularizer based on the image gradient. It enforces low regularization smoothness along image edges and high smoothness in homogeneous image regions.

We denote ADT by G following [14]. ADT is a pixel-wise tensor derived based on the gradient of the input image. Using an image I and normalized direction of the image gradient $\mathbf{n} = \frac{\nabla I}{|\nabla I|}$, pixel-wise ADT $G \in \mathbb{R}^{2\times 2}$ is derived as follows.

$$G = \exp\left(-a\left|\nabla I\right|^{b}\right)\mathbf{n}\mathbf{n}^{\mathrm{T}} + \mathbf{n}^{\perp}\mathbf{n}^{\perp\mathrm{T}} \qquad (1)$$

Here, $\mathbf{n}^{\perp}$ is the normal vector to the gradient and the scalars a and b are hyper parameters to adjust the magnitude and the sharpness of the tensor. Note that G is quadratic to $\mathbf{n}^{\perp}$ in Eq. (1), and hence is not dependent on the choice of normal vector $\mathbf{n}^{\perp}$ out of two.

ADT has been used to anisotropically weight the TGV regularization first proposed by Bredies et al. in [15]. TGV is composed of polynomials of arbitrary order, which allows us to reconstruct piecewise polynomial functions.

In [2], the total energy for depth completion is defined as the combination of the data term $C(u)$ and the second order

TGV term $R(u, \mathbf{v})$ with ADT. With the upsampled depth map $u : \Omega \to \mathbb{R}$ and the relaxation variable $\mathbf{v} : \Omega \to \mathbb{R}^2$, this energy is defined as:

$$\int_{\Omega} C(u) + R(u, \mathbf{v}) d\mathbf{x} \tag{2}$$

where, the data term $C(u)$ and TGV term $R(u, \mathbf{v})$ are expressed as:

$$C(u) = \lambda_d w \left| u - d \right|^2 \tag{3}$$

$$R(u, \mathbf{v}) = \lambda_s \left| G \left(\nabla u - \mathbf{v} \right) \right| + \lambda_a \left| \nabla \mathbf{v} \right| \tag{4}$$

Here, $w \in \mathbb{R}$ is the pixel-wise weight for the data term, and hyper parameters $\lambda_s \in \mathbb{R}$, $\lambda_a \in \mathbb{R}$, and $\lambda_d \in \mathbb{R}$ are weights for each of the energy terms. d and w are zero at pixels where the depth is not captured.

The energy of Eq. (2) is convex and can be optimized to a global minimum through the primal dual algorithm [16]. The energy minimization allows ∇u to be smooth on Ω through the relaxation variable $\mathbf{v}$, while constraining the value u around d. The upsampled depth map is derived as the minimizer of the total energy.

B. Binary Anisotropic Diffusion Tensor

Although ADT can control the magnitude of smoothness, it is not sufficient to discretize multiple objects in a scene. ADT is continuous and still applies weak smoothness constraint at the occlusion boundaries. The effect is visually apparent in point cloud data as surface-like artifacts connecting discrete objects.

We propose B-ADT to overcome this problem. B-ADT is an extension of ADT that can eliminate smoothness constraint anisotropically. As the name suggests, B-ADT is composed of only 0 and 1.

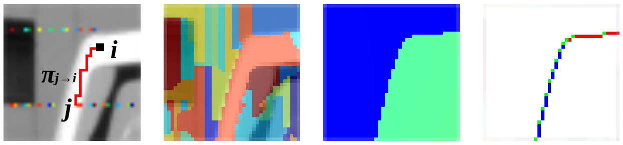
B-ADT is derived as extremes of ADT. We consider two extremes where the magnitude of the image gradient $|\nabla I|$ is either extremely small or extremely large. More importantly, when $|\nabla I|$ is extremely large, it can totally eliminate the smoothness constraint in the TGV term. Following the definition of ADT in Eq. (1), we can derive the extremes as follows.

$$\lim_{|\nabla I| \to +0} G = \mathbf{n}\mathbf{n}^{\mathrm{T}} + \mathbf{n}^{\perp}\mathbf{n}^{\perp\mathrm{T}} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{5}$$

$$\lim_{|\nabla I| \to +\infty} G = \mathbf{n}^{\perp}\mathbf{n}^{\perp\mathrm{T}} \tag{6}$$

We use these extremes as B-ADT according to the locations and directions of occlusion boundaries. Occlusion boundaries are between objects which are not connected to each other and the depth is discontinuous across them. Note, on the other hand, that the depth is smooth by crossing connected boundaries where different objects are in contact. We consider the extreme of ADT with $|\nabla I| \to +\infty$ on occlusion boundaries, and $|\nabla I| \to +0$ otherwise.

For each pixel, the value of B-ADT is decided based on two conditions, "A: the pixel is on a vertical occlusion boundary" and "B: the pixel is on a horizontal occlusion boundary". Here, a vertical occlusion boundary is a vertical line segment across which the depth is discontinuous in the horizontal direction. Accordingly, a horizontal occlusion boundary is a horizontal line segment across which the depth is discontinuous in the vertical direction.



(a) Path (b) Voronoi cells (c) Depth map (d) Boundary

Fig. 3. Illustration of IGNNS and the boundary derivation. (a) Pixels with the LIDAR depth are indicated with colors. The path cost is the weighted sum of the length of the path and accumulation of the norm of the image gradient along the path. (b) Pixels masked by the same color have the same nearest neighbor. (c) The piecewise constant depth map is filled with depth value at the nearest neighbor of each pixel. (d) The occlusion boundaries are derived by thresholding the norm of the gradient of the piecewise constant depth map. The color scheme in the figure is the same as Fig. 2 (c)

Because the image is defined on 2D grid, every pixel is classified as one of 4 types: "neither A nor B ($\neg A \wedge \neg B$)", "A but not B ($A \wedge \neg B$)", "not A but B ($\neg A \wedge B$)", and "A and B ($A \wedge B$)". In Fig. 2 (c) and Fig 3 (d), we denote the pixels by $\neg A \wedge \neg B$ as white, $A \wedge \neg B$ as blue, $\neg A \wedge B$ as red, and $A \wedge B$ as green. If we let $\mathbf{n}$ to be defined as $\mathbf{e}_x$ on vertical and $\mathbf{e}_y$ on horizontal occlusion boundaries, we can write the B-ADT $\bar{G}$ for each condition as follows.

$$\bar{G}_{\neg A \wedge \neg B} = \lim_{|\nabla I| \to +0} G = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{7}$$

$$\bar{G}_{A \wedge \neg B} = \lim_{|\nabla I| \to +\infty, \mathbf{n}=\mathbf{e}_x} G = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \tag{8}$$

$$\bar{G}_{\neg A \wedge B} = \lim_{|\nabla I| \to +\infty, \mathbf{n}=\mathbf{e}_y} G = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{9}$$

In case of $A \wedge B$, we set B-ADT as product of $\bar{G}_{A \wedge \neg B}$ and $\bar{G}_{\neg A \wedge B}$, so that it applies discontinuity in both directions. As a result, we have:

$$\bar{G}_{A \wedge B} = \bar{G}_{A \wedge \neg B} \bar{G}_{\neg A \wedge B} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \tag{10}$$

We can verify the effects of B-ADT by applying it to the TGV term in Eq. (4). By substituting ADT G with B-ADT $\bar{G}$ in Eq. (4), we have our TGV term $\bar{R}(u, \mathbf{v})$, with denoting $\mathbf{v} = (v_x, v_y)^T$, as:

$$\bar{R}(u, \mathbf{v}) = \begin{cases} \lambda_s \left| \nabla u - \mathbf{v} \right| + \lambda_a \left| \nabla \mathbf{v} \right| & \text{if } \neg A \wedge \neg B \\ \lambda_s \left| \frac{\partial u}{\partial y} - v_y \right| + \lambda_a \left| \nabla \mathbf{v} \right| & \text{if } A \wedge \neg B \\ \lambda_s \left| \frac{\partial u}{\partial x} - v_x \right| + \lambda_a \left| \nabla \mathbf{v} \right| & \text{if } \neg A \wedge B \\ \lambda_a \left| \nabla \mathbf{v} \right| & \text{if } A \wedge B \end{cases} \tag{11}$$

From the above equations, we can see that there is no smoothness cost on depth across occlusion boundaries. For example, x directional derivative of depth u does not contribute to the term at all on vertical boundaries. Hence, an extremely large value of a gradient of u can be taken across the boundaries. As a consequence, discontinuous changes are allowed across the boundaries with B-ADT.

C. Image-guided Nearest Neighbor Search

To derive B-ADT, we need occlusion boundaries well correlated to the depth maps. For that, we perform IGNNS which searches the shortest path in terms of the accumulation of the image gradient. IGNNS is based on the observation that two points tend to be on the same object if there is a path between them which does not cross a lot of edges on the image. The result of IGNNS is also used in our depth completion energy as we see in Section III-E. We illustrate IGNNS in Fig. 3 (a) - (c).

IGNNS outputs a piecewise constant depth map from an input sparse depth map and an image. Let D be the set of pixels where the input sparse depth map d has a value and $\mathbf{i}_* \in D$ as the nearest neighbor of $\mathbf{i} \in \Omega$. Considering the nearest neighbors of every pixel in the image domain, we get a piecewise constant dense depth map $\bar{d} : \Omega \to \mathbb{R}$ as

$$\bar{d}(\mathbf{i}) = d(\mathbf{i}_*). \tag{12}$$

IGNNS is a way to derive $\mathbf{i}_*$ with guidance of the input image I. Let $\pi_{\mathbf{j}\to\mathbf{i}}$ be a path from pixel $\mathbf{j}$ to pixel $\mathbf{i}$ on the image grid. Here, $\pi_{\mathbf{j}\to\mathbf{i}}$ is expressed as set of pixels on the path. In IGNNS, we search the nearest neighbor $\mathbf{i}_*$ from $\mathbf{i}$ as:

$$\mathbf{i}_* = \underset{\mathbf{j}\in\mathbf{D}}{\operatorname{argmin}} \min_{\pi_{\mathbf{j}\to\mathbf{i}}} \{ \sum_{\mathbf{k}\in\pi_{\mathbf{j}\to\mathbf{i}}} \|\nabla I(\mathbf{k})\|^2 + c|\pi_{\mathbf{j}\to\mathbf{i}}| \} \tag{13}$$

where the hyper parameter c is the constant cost of the unit path length and $|\pi_{\mathbf{j}\to\mathbf{i}}|$ is the number of pixels in $\pi_{\mathbf{j}\to\mathbf{i}}$.

With the nearest neighbor $\mathbf{i}_*$ of Eq (13), we derive the piecewise constant depth map $\bar{d}$ using Eq. (12).

D. Boundary derivation

We derive the occlusion boundaries based on IGNNS result $\bar{d}$. With pre-defined threshold t, we decide a pixel $\mathbf{x}$ to be on a vertical occlusion boundary if $|\frac{\partial \bar{d}(\mathbf{x})}{\partial x}| > t$ and to be on a horizontal occlusion boundary if $|\frac{\partial \bar{d}(\mathbf{x})}{\partial y}| > t$ (Fig. 3 (d)).

This thresholding can generate false occlusion boundaries especially on the ground since it is often a large plane parallel to the view direction and has a wide range of depth. Hence, we filter out occlusion boundaries detected on the ground. We can generate efficient ground labels from an input sparse depth map as follows.

We convert the input sparse depth map to a point cloud and apply RANSAC plane model segmentation to it. Then, we label points in and below the detected plane as ground. Finally, we label each pixel by the label of its IGNNS nearest neighbor. This process has two hyper parameters regarding RANSAC plane segmentation: the number of iterations N_{ransac} and distance threshold t_{ransac}. Our implementation uses Point Cloud Library [17] for the plane segmentation.

In comparison to conventional boundary detection methods relying only on an image as [18], our method jointly uses an image and a depth map. Hence the boundaries are well correlated to depth maps and suitable for B-ADT derivation.

E. Energy Minimization

We derive the upsampled depth maps by minimizing the energy with B-ADT weighted TGV regularization. Our energy minimization follows that of ADT aided depth completion [2] with two major differences. First, we apply B-ADT instead of ADT. Second, we use a densified depth map from raw LIDAR measurements, resulting from IGNNS, for the data term instead of the depth camera as used in [2]. We used a densified data $\bar{d}$ because we found, through our experiment, that the variational method was very difficult to stably converge using sparse data (see Appendix I). Moreover, we use inverse depth $\bar{d}^{-1}$ instead of depth $\bar{d}$ to balance the contribution of near and far depth as was done in [19]. To further help the convergence, we scale the inverse depth so that the maximum of $\bar{d}^{-1}$ is one.

We define our data term and TGV term for the energy as:

$$\bar{C}(u) = \lambda_d w \left| u - \bar{d}^{-1} \right|^2 \tag{14}$$

$$\bar{R}(u, \mathbf{v}) = \lambda_s \left| \bar{G} \left(\nabla u - \mathbf{v} \right) \right| + \lambda_a \left| \nabla \mathbf{v} \right| \tag{15}$$

The depth completion becomes the following energy minimization problem.

$$\min_{u,\mathbf{v}} \int_\Omega \bar{C}(u) + \bar{R}(u, \mathbf{v}) d\mathbf{x} \tag{16}$$

We minimize the above energy using the primal dual algorithm [16]. First, we discretize the energy and conduct the Legendre-Fenchel transform of TGV term, while introducing dual variables $\mathbf{p} : \Omega \to \mathbb{R}^2$ and $\mathbf{q} : \Omega \to \mathbb{R}^4$. Then $\int \bar{R}(u, \mathbf{v}) d\mathbf{x}$ becomes as:

$$\max_{\|\mathbf{p}\|_\infty \le \lambda_s} \langle \bar{G} \left(\nabla u - \mathbf{v} \right), \mathbf{p} \rangle + \max_{\|\mathbf{q}\|_\infty \le \lambda_a} \langle \nabla \mathbf{v}, \mathbf{q} \rangle. \tag{17}$$

Then we apply gradient descent on u, $\mathbf{v}$ and gradient ascent on $\mathbf{p}$, $\mathbf{q}$ to derive the primal dual iterations for the energy minimization as

$$\mathbf{p}^{n+1} = \Pi_P \left(\mathbf{p}^n + \tau_p \bar{G} \left(\nabla \hat{u}^n - \hat{\mathbf{v}}^n \right) \right) \tag{18}$$

$$\mathbf{q}^{n+1} = \Pi_Q \left(\mathbf{q}^n + \tau_q \nabla \hat{\mathbf{v}}^n \right) \tag{19}$$

$$u^{n+1} = \frac{u^n + \tau_u \left(\operatorname{div} \bar{G} \mathbf{p}^{n+1} + \lambda_d w \bar{d}^{-1} \right)}{1 + \tau_u \lambda_d w} \tag{20}$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \tau_v \left(\mathbf{p}^{n+1} + \operatorname{div} \mathbf{q}^{n+1} \right) \tag{21}$$

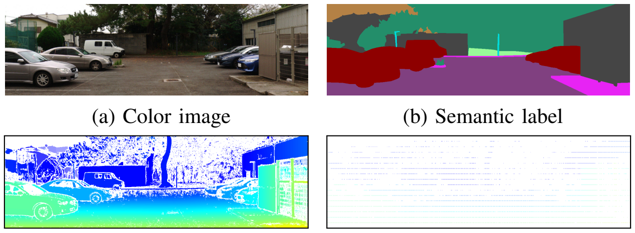
$$\hat{u}^{n+1} = 2u^{n+1} - u^n \tag{22}$$

$$\hat{\mathbf{v}}^{n+1} = 2\mathbf{v}^{n+1} - \mathbf{v}^n, \tag{23}$$

where hyper parameters τ_p, τ_q, τ_u, and τ_v are step sizes of iterations, and the proximal mappings are given as $\Pi_P(\mathbf{p}) = \frac{\mathbf{p}}{\max\{1, \|\mathbf{p}\|/\lambda_s\}}$ and $\Pi_Q(\mathbf{q}) = \frac{\mathbf{q}}{\max\{1, \|\mathbf{q}\|/\lambda_a\}}$. For initialization, we set $u^0 = \bar{d}^{-1}$, $\mathbf{v}^0 = 0$, $\mathbf{p}^0 = 0$, $\mathbf{q}^0 = 0$, $\hat{u}^0 = \bar{d}^{-1}$, $\hat{v}^0 = 0$. This optimization efficiently runs in parallel and can run in real time on modern GPUs.

IV. Evaluation

In this section, we evaluate our depth completion method on multiple aspects: depth map accuracy, processing time, point cloud quality, and parameter sensitivity.



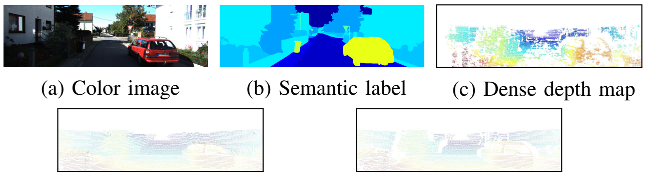
(a) Color image

(b) Semantic label

(c) Dense depth map

(d) Sparse depth map

Fig. 4. A frame of Komaba. In depth maps, white indicates no value in data.



(a) Color image

(b) Semantic label

(c) Dense depth map

(d) Sparse depth map

(e) Filtered sparse depth map

Fig. 5. A frame of KITTI. In depth maps, white indicates no value in data.

A. *Datasets*

We evaluated our method on two datasets, Komaba (Fig. 4) and KITTI (Fig 5). For Komaba, we used 5 frames (Frame 8, 12, 18, 29, 35) following the evaluation of [7]. For KITTI, we used 97 frames which are with semantic labeling by [20]. Both datasets have colored images, sparse depth maps, dense depth maps, and manual semantic labels.

Because of the large disparity between the LIDAR and the camera, depth maps in KITTI contain occluded background noises. Since our method does not handle such noises, we pre-processed the sparse depth maps to remove them (Fig. 5 (e)). We used the filtered depth maps as input in the experiments on KITTI.

More details about the datasets and our occluded background noise removal approach are described in Appendix II.

B. *Experimental Conditions*

a) Baseline methods: For the nearest neighbor search, we compared our method with two baselines. One is the nearest neighbor search on distance only (NNS). The other is Joint Nearest Neighbor Search (JNNS), which is based on Joint Distance (JD) of color and space. JD is based on the metric for the interpolation of Kopf et al. [9]. JD from $\mathbf{i} \in \Omega$ to $\mathbf{j} \in \Omega$ is defined with weights α and β as

$$\mathrm{JD}(\mathbf{i}, \mathbf{j}) = \alpha \left\| \mathbf{i} - \mathbf{j} \right\|^2 + \beta \left(I(\mathbf{i}) - I(\mathbf{j}) \right)^2 . \tag{24}$$

For depth completion, we compared our methods with unsupervised methods of Kopf et al. [9] and Ferstl et al. [2] on both datasets, unsupervised method of Ku et al.'s [10] on KITTI only, and semi-supervised method of Hirata et al. [2] on Komaba only. Additionally, we improved Ferstl et al.'s method [2] by using our data term $\bar{C}(u)$ instead of their $C(u)$ ("IGNNS+ADT").

We experimented with three versions of our method in terms of the use of the ground labels. The first is without occlusion boundary filtering ("Ours no label"). The second is with occlusion boundary filtering using the generated ground labels ("Ours"). Refer Section III-D for our label generation. The third is with occlusion boundary filtering using the manual labels in the datasets ("Ours man-label"). Since "Ours man-label" uses additional inputs, the results are reference only and not used in evaluations.

b) Hyper parameters: We used the following parameters for our method on both datasets. $c = 0.01$ for IGNNS, $t = 2.0$ [m] (meters) for the boundary derivation, and $\lambda_s = 0.2$, $\lambda_a = 1.6$, $\lambda_d = 0.2$, $w = \bar{d}$, $\tau_p = \tau_q = 1.0/\sqrt{8}$, $\tau_u = \tau_v = 1.0/\sqrt{12}$ for the energy minimization. $N_{\mathrm{ransac}} = 1000$ and $t_{\mathrm{ransac}} = 0.2$ [m] for the ground detection. The number of iterations is 400 for Komaba and 200 for KITTI. This is because it took more iterations to converge for Komaba since the initial depth maps are sparser.

We used the following parameters on baseline methods. For JNN and Kopf et al's method [9], we used $\alpha = 100$, $\beta = 0.2$. For ADT aided methods, based on our tuning, we used $a = 5.0$, $b = 0.5$ for [2], and $a = 10.0$, $b = 0.5$ for "IGNNS+ADT", and the same parameters as our method for the energy minimization. For Hirata et al.'s method [7], we used the manual semantic labels and motion stereo provided in the dataset and conducted parameter tuning to adjust to our sampling. Finally, for Ku et al's method [10], we used their publicly available implementation as is, which was tuned for the KITTI depth completion benchmark by the authors.

Additionally, we used the following parameters for the pre-processing on KITTI. $r_{\mathrm{occ}} = 256.0 * d^{-1}$ and $t_{\mathrm{occ}} = 2.0$ [m] for the occluded background filtering (see Appendix II).

C. *Depth Map Accuracy*

We evaluated the accuracy by Mean Absolute Error (MAE) between the resulting depth maps and the ground truths in the datasets. We show the nearest neighbor search results in Tables I and II and the depth completion results in Tables III and IV.

In Tables I and II, we see IGNNS outperformed NNS and JNNS on both datasets. In Tables III and IV, we see our depth completion outperformed the previous and baseline methods with no ground label on both datasets (see "Ours no label"). Moreover, occlusion boundary filtering by our generated ground labels improved the accuracy further (see "Ours").

In Tables III and IV, we see "Ours" is competitive to "Ours man-label". The MAE differences between them are less than 5 [mm] on both datasets. Furthermore, "Ours" outperformed "Ours man-label" on KITTI. This result shows that the generated ground labels were as effective as the manual labels for our purpose.

D. *Processing time*

We implemented our method on an Ubuntu 18.04 LTS laptop computer running Intel(R) Core(TM) i9-8950HK @

TABLE I

Nearest neighbor search results on Komaba dataset

	MAE [mm]					
	F. 8	F. 12	F. 18	F. 29	F. 35	Avg.
NNS	265.1	545.2	167.3	160.6	234.1	274.5
JNNS	**253.2**	546.8	168.2	160.5	246.4	275.0
IGNNS	267.1	**433.5**	**164.5**	**115.7**	**226.8**	**241.5**

TABLE II

Nearest neighbor search results on KITTI dataset

	MAE [mm]
NNS	530.3
JNNS	527.9
IGNNS	**526.1**

2.90GHz × 6 cores and GeForce RTX 2080 GPU. On average, the processing time per frame was 0.163 (IGNNS: 0.025, the ground label generation: 0.032, the energy minimization 0.107) second on Komaba and 0.120 (IGNNS: 0.023, the ground label generation: 0.070, the energy minimization 0.027) second on KITTI datasets. The results indicate that it can process 6 - 8 frames per second on a single thread and can be applied to real-time applications with proper parallelization. Longer processing time for the energy minimization on Komaba is because the number of iterations is larger than on KITTI: 400 v.s. 200. Ground detection took longer on KITTI than on Komaba because the input depth maps have more data.

E. Point cloud quality

We show selected results as depth maps and point clouds in Fig. 6. The advantage of our method is more obvious in point clouds than in depth maps. In Fig. 6, we can see that only our method preserves the discontinuity at occlusion boundaries. Other methods generate unnatural artifacts between foreground and background objects because of the smoothness imposed at the boundary during interpolation. We can also see the effect in the error maps. Results from other methods show gradual changes in the error around object boundaries, whereas our results show immediate changes.

F. Parameter Sensitivity

Our method introduces two hyper parameters: c for IGNNS and t for the boundary derivation. We conducted parameter sensitivity evaluation on these. Here, we used "Ours no label" as a baseline to see the effects of the variables directly, and the other variables are fixed to values in Section IV-B.

In Fig. 7, we plot the average MAE on Komaba dataset with various values of c or t. For c, MAE was the minimum (239.3) at $c = 0.04$, and was less than or equal to the result of IGNNS in Table I when $0.01 \leq c \leq 0.055$. For t, MAE was the minimum (217.7) at $t = 3.0$, and was less than or equal to the result of "Ours no label" in Table III when $2.0 \leq t \leq 4.5$.

TABLE III

Depth completion results on Komaba dataset

	MAE[mm]					
	F. 8	F. 12	F. 18	F. 29	F. 35	Avg.
Kopf et al. [9]	291.9	533.6	163.0	200.2	383.3	314.4
Ferstl et al. [2]	603.9	1045.4	301.0	660.4	1094.1	740.9
Hirata et al. [7]	1036.8	1096.2	421.7	443.2	711.9	742.0
IGNNS+ADT	251.7	397.5	159.2	131.0	246.5	237.2
Ours no label	255.9	383.8	135.8	**116.7**	**216.6**	221.8
Ours	**227.7**	**321.4**	**134.3**	127.0	221.6	**206.4**
Ours man-label	228.7	321.3	134.0	116.9	213.7	202.9

TABLE IV

Depth completion results on KITTI dataset

	MAE [mm]
Kopf et al. [9]	744.1
Ferstl et al. [2]	928.7
Ku et al. [10]	614.2
IGNNS+ADT	525.3
Ours no label	506.5
Ours	**500.2**
Ours man-label	504.1

Although good parameters are data-dependent, our observation suggests the following for reasonable parameter choices. Set c to be on the scale of $1/100$ to $1/10$ against the square of the maximum color value, which in our case is 1.0. Set t to be on the scale of distance between the foreground and the background of the scene, e.g. several meters for the outdoor environments and several centimeters for the desktop environment.

V. Benchmark Evaluation

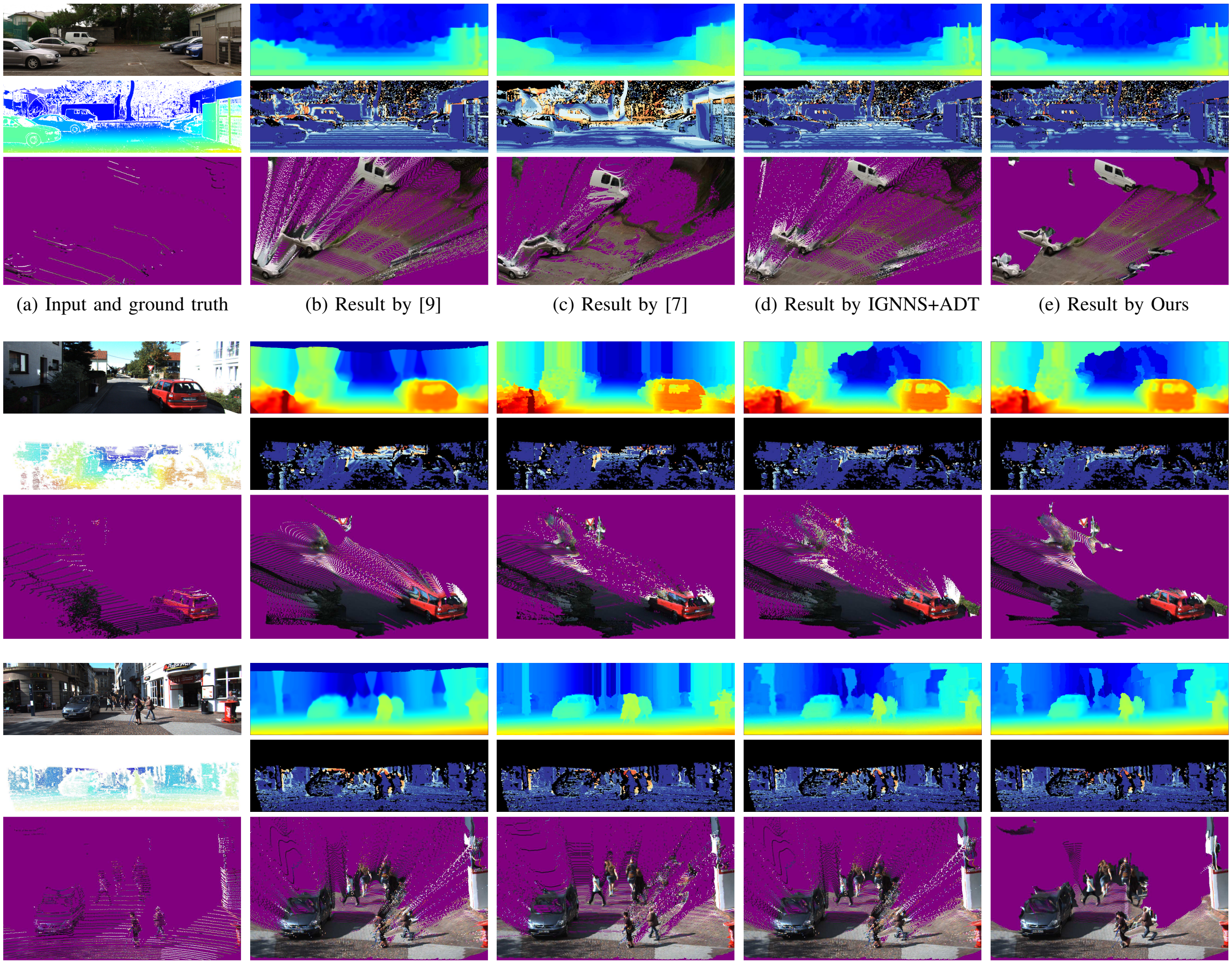
We applied our method to the public KITTI depth completion benchmark [3] to compare our method with various methods. In the benchmark, there are the validation data for the tuning and the test data for the evaluation. Both of them contain 1,000 pairs of an image and a sparse depth map. The parameters used in the benchmark are the same as "Ours" in Section IV-B except for the following two parameters: $w = \bar{d}^{2.5}$ in the energy minimization and $r_{\text{occ}} = 96.0 * d^{-1}$ in the pre-processing.

In Table V, we show our results with all unsupervised [10] and DNN enhanced self-supervised [5], [8] methods listed in the benchmark website at the time of writing. The results on the validation data are based on their published papers. Among them, ours performed the best in terms of MAE.

VI. Conclusion

We proposed B-ADT and its application to depth completion with IGNNS. Our method is fully unsupervised, runs in real time, and generates a depth map discontinuous at occlusion boundaries and smooth elsewhere, which is suitable for point cloud representation.

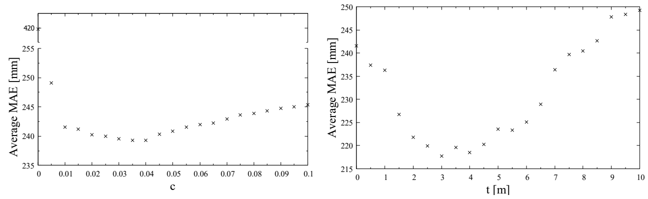
We showed that our method outperformed both existing and baseline methods in terms of the accuracy of the depth maps and the visual quality of the point clouds. Among the methods we evaluated, our method is the only one to preserve the discontinuity between foreground and background objects.



(a) Input and ground truth (b) Result by [9] (c) Result by [7] (d) Result by IGNNS+ADT (e) Result by Ours

(f) Input and ground truth (g) Result by [9] (h) Result by [10] (i) Result by IGNNS+ADT (j) Result by Ours

Fig. 6. (a) - (e) Results on a selected frame of Komaba. (f) - (j) Results on two selected frames of KITTI. (a), (f) From top to bottom: the input image, the ground truth depth map, and the input LIDAR in point cloud. (b) - (e), (g) - (j) From top to bottom: the depth completion result, the error map, and the depth completion result in point cloud. By comparing the resulting point clouds, we see only our method preserves the discontinuity at occlusion boundaries.



(a) Sesitivity to c (b) Sensitivity to t

Fig. 7. Parameter sensitivity results on the Komaba dataset. MAE is calculated (a) on IGNNS results (b) after energy minimization.

TABLE V

RESULTS ON THE BENCHMARK

	MAE [mm]	
	validation	test
Ma et al. [5]	358.9	350.3
Ku et al. [10]	305.3	302.6
Wong et al. [8]	305.1	299.4
Ours	**300.7**	**298.7**

In this paper, we focused on B-ADT for depth completion. However, B-ADT is a general idea that enables discontinuous but smooth optimization in variational energy minimization. Variational methods have been applied to many tasks such as stereo matching, optical flow estimation, segmentation, and so on. We believe it will be interesting to explore the possibility of B-ADT in many applications in the future.

REFERENCES

[1] 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.

[2] D. Ferstl, 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.

[3] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *International Conference on 3D Vision (3DV)*. IEEE, 2017, pp. 11–20.

[4] Z. Huang, J. Fan, S. Cheng, S. Yi, X. Wang, and H. Li, "Hms-net: Hierarchical multi-scale sparsity-invariant network for sparse depth completion," *IEEE Transactions on Image Processing*, 2019.

[5] 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.

[6] 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.

[7] A. Hirata, R. Ishikawa, M. Roxas, and T. Oishi, "Real-time dense depth estimation using semantically-guided lidar data propagation and motion stereo," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3806–3811, 2019.

[8] 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.

[9] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," in *ACM Transactions on Graphics (ToG)*, vol. 26, no. 3. ACM, 2007, p. 96.

[10] J. Ku, A. Harakeh, and S. L. Waslander, "In defense of classical image processing: Fast depth completion on the cpu," in *Conference on Computer and Robot Vision (CRV)*. IEEE, 2018, pp. 16–22.

[11] 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.

[12] L. Chen, Y. He, J. Chen, Q. Li, and Q. Zou, "Transforming a 3-d lidar point cloud into a 2-d dense depth map through a parameter self-adaptive framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 165–176, 2016.

[13] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof, "Anisotropic huber-l1 optical flow." in *The British Machine Vision Conference (BMVC)*, vol. 1, no. 2, 2009, p. 3.

[14] G. Kuschk and D. Cremers, "Fast and accurate large-scale stereo reconstruction using variational methods," in *IEEE International Conference on Computer Vision Workshops*, 2013, pp. 700–707.

[15] K. Bredies, K. Kunisch, and T. Pock, "Total generalized variation," *SIAM Journal on Imaging Sciences*, vol. 3, no. 3, pp. 492–526, 2010.

[16] T. Pock, L. Zebedin, and H. Bischof, "Tgv-fusion," in *Rainbow of computer science*. Springer, 2011, pp. 245–258.

[17] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[18] K. Murasaki, K. Sudo, and Y. Taniguchi, "Occlusion boundary detection based on mid-level figure/ground assignment features," in *International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 4707–4711.

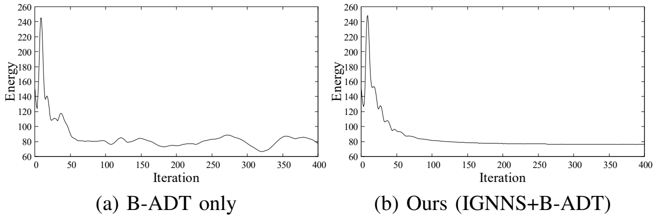
[19] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *International Conference on Computer Vision (ICCV)*. IEEE, 2011, pp. 2320–2327.

[20] P. Xu, F. Davoine, J.-B. Bordes, H. Zhao, and T. Denœux, "Multimodal information fusion for urban scene understanding," *Machine Vision and Applications*, vol. 27, no. 3, pp. 331–349, 2016.

Appendix I
Convergence study

We experimentally found that the energy minimization is difficult to converge stably with the sparse data term. We compared the convergence of the energy minimization between our method ("IGNNS+B-ADT") and the baseline method ("B-ADT only"). Our method and the baseline method differ only in the data term of the energy. The data term of "B-ADT only" is based on the input sparse depth d. We used the inverse of the depth as in our method. Hence, the data term of baseline method $C'(u)$ is defined as the following.

$$C'(u) = \lambda_d w \left| u - d^{-1} \right|^2 \tag{25}$$



(a) B-ADT only

(b) Ours (IGNNS+B-ADT)

Fig. 8. Convergence of the energy. (a) With the sparse data term, the energy oscillates through the iterations and does not converge stably. (b)With the dense data term, the energy converges stably.

With the same hyper parameters as in Section IV-B, we observed the change in the energy during its minimization on Frame 12 of the Komaba dataset (Fig. 8). We can see that the energy of our method is converging stably in contrast to the energy of "B-ADT only" which oscillates through the iteration. Note that the two methods have different energy and there is no point in comparing the values of the energy.

Appendix II
Datasets details

In this appendix, we explain the datasets introduced in Section IV-A in more detail. Note KITTI in this section is different from the benchmark dataset in Section V.

Komaba is used in [7] and publicly available[1]. There are 56 image pairs with resolutions of 1238 × 374, dense depth maps captured with Faro Focus S150[2], simulated LIDAR data, semantic segmentation results, motion stereo, and 7 manual segmentation labels. We selected 5 frames (Frame 8, 12, 18, 29, 35) which are the same as frames used in the evaluation of [7]. And we sampled dense depth maps with specifications of Velodyne VLP-16[3]: vertical resolution 1.3 degree, horizontal resolution 0.2 degree, receptive range from 0.5 - 100 [m].

KITTI is introduced by [1] and widely used. We used manual semantic labels from Xu et al. [20]. And we collected corresponding dense depth maps from the depth completion benchmark dataset [3]. In this way, we collected 97 frames of dataset composed of scenes of 72 City, 14 Residential, 4 Campus, and 7 Road.

In KITTI, projected LIDAR depth maps contain the depth of the occluded background because of the large disparity between LIDAR and cameras. We removed them by a simple pre-processing. For every measured depth d, we draw a lower semi-circle with radius r_{occ} centered at the position of d, and remove depth if it is covered by the semi-circle and its value is farther than threshold t_{occ} comparing with the current semi-circle center. Our pre-processing is based on the observation that occluded background depth locates at below of foreground object depth on the image plane.

[1] https://github.com/menandro/upsampling

[2] https://www.faro.com/

[3] https://velodyneLIDAR.com/