2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

PLG-IN: Pluggable Geometric Consistency Loss with Wasserstein Distance in Monocular Depth Estimation

Noriaki Hirose[1], Satoshi Koide[1], Keisuke Kawano[1] and Ruho Kondo[1]

Abstract— **We propose a novel objective for penalizing geometric inconsistencies and improving the depth and pose estimation performance of monocular camera images. Our objective is designed using the Wasserstein distance between two point clouds, estimated from images with different camera poses. The Wasserstein distance can impose a soft and symmetric coupling between two point clouds, which suitably maintains geometric constraints and results in a differentiable objective. By adding our objective to those of other state-of-the-art methods, we can effectively penalize geometric inconsistencies and obtain highly accurate depth and pose estimations. Our proposed method was evaluated using the KITTI dataset.**

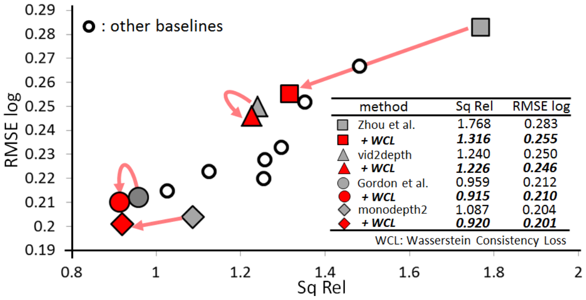
I. INTRODUCTION

Understanding the three-dimensional (3D) structures of environments and objects is important for the navigation of autonomous vehicles and robotic manipulation [1], [2]. In recent years, depth estimation using RGB monocular images has been actively studied owing to the popularity of deep learning. Cameras can function as inexpensive and affordable sensors for autonomous vehicles and robots [3], [4]. Hence, they can be viable alternatives to expensive LiDARs. However, it is difficult to obtain a considerable number of depth image ground truths from LiDAR data, for application to machine learning techniques, and data collection is itself a challenge.

Self-supervised learning using video clips captured by robots and vehicles enables the learning of depth and pose estimation networks without ground truth depth images and poses. Here, pose estimation refers to the relative camera pose estimation between two consecutive images. The image reconstruction loss proposed by Zhou et al. [5] has a significantly improved accuracy. Various pluggable objectives [6], [7], [8], [9], network structures, data augmentation methods, and masking methods for dynamic and occluded objects [10], [11] have also been suggested to enable various improvements [5], [12], [13], [14].

In this paper, we propose a novel pluggable objective to evaluate geometric inconsistencies with respect to these prior methods. Because the geometry of objects or environments does not depend on the camera pose, the two sets of 3D point clouds estimated in the same coordinates from different camera poses should be consistent.

Several previous studies have proposed various objectives that attempt to penalize inconsistencies in 3D geometric constraints [7], [12], [9], [15], [16]. However, their performance is limited due to a geometrically inconsistent evaluation with an approximation, geometrically inconsistent coupling between 3D point clouds, and dependence on other undifferentiable algorithms to obtain the coupling [7], [12], [9], [15], [16].



	method	Sq Rel	RMSE log
■	Zhou et al.	1.768	0.283
■	***+ WCL***	***1.316***	***0.255***
▲	vid2depth	1.240	0.250
▲	***+ WCL***	***1.226***	***0.246***
●	Gordon et al.	0.959	0.212
●	***+ WCL***	***0.915***	***0.210***
◆	monodepth2	1.087	0.204
◆	***+ WCL***	***0.920***	***0.201***

WCL: Wasserstein Consistency Loss

Fig. 1: **Evaluation of monocular depth estimation, with and without our proposed Wasserstein consistency loss (WCL) on the KITTI raw dataset using an image pixel resolution of 416×128.** Red markers are the results of the WCL. Gray markers are the baselines without the WCL. Our WCL can enhance the performance of monocular depth estimation. It should be noted that a smaller Sq Rel and a smaller RMSE log are better.

In this study, we address these issues to improve the accuracy of a depth and pose estimation. We propose a novel objective inspired by recent successful studies on the Wasserstein distance, including generative modeling [17], embedding [18], [19], and domain adaptation [20]. The proposed method can measure the consistency between two point clouds and impose a penalty to achieve a more accurate estimation. In contrast to the baselines, our method attempts to measure the geometric consistency from 3D point clouds without any indirect processing or bold approximation. In addition, the mathematical formulation of our objective is smooth and symmetric, which is advantageous for an efficient and effective training process.

The major contributions of this study are 1) the proposal of a novel objective, i.e., the Wasserstein consistency loss (WCL), to evaluate the consistency in 3D geometric constraints, and 2) a comparative evaluation to demonstrate that the accuracy of several state-of-the-art approaches is improved when "plugging in" the WCL as shown in Fig. 1. To the best of our knowledge, our study is the first application using the Wasserstein distance for depth and pose estimation. We quantitatively and qualitatively demonstrate the benefits of our WCL using the KITTI dataset.

II. RELATED STUDIES

a) Self-supervised monocular depth estimation: Self-supervised depth estimation has recently become popularized

[1] Noriaki Hirose et al. are with TOYOTA Central R&D Labs., INC., Japan hirose@tytlabs.co.jp

978-1-7281-9077-8/21/$31.00 ©2021 IEEE

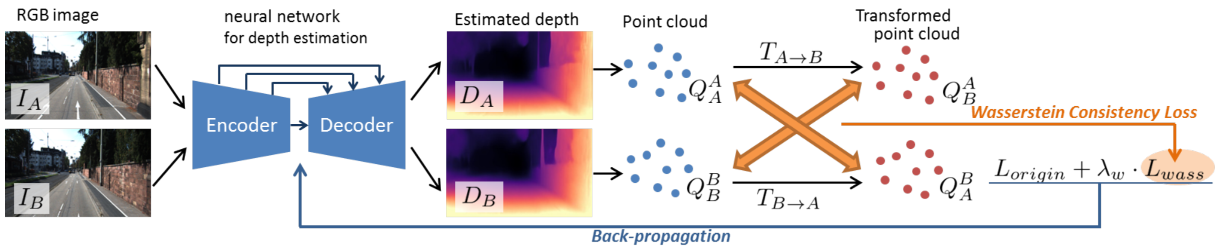


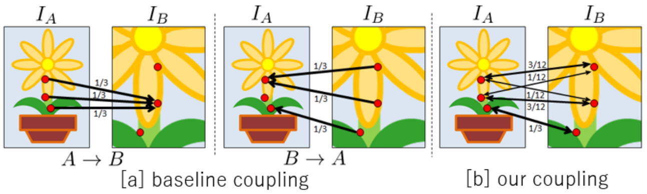
Fig. 2: **Overview of our proposed approach.** We feed RGB images I_A and I_B into a neural network to estimate the depth images D_A and D_B, respectively. From D_A and D_B, we obtain the point clouds Q^A_A and Q^B_B, and Q^B_A and Q^A_B, respectively, by applying the intrinsic camera parameters and the estimated transformation matrices, $T_{A\rightarrow B}$ and $T_{B\rightarrow A}$. To penalize a geometric inconsistency, we propose the addition of the WCL L_{wass} to the original cost functions, L_{origin}, of recent state-of-the-art approaches. This is done to train the neural networks.

[21], [22], [23], [24], [25], [26], [27]. Zhou et al. [5] and Vijayanarasimhan et al. [28] demonstrated one of the first approaches for self-supervised monocular depth and pose estimation. Their approach simultaneously trained the model to estimate the depth and pose based on knowledge regarding structure from motion (SfM). Garg et al. [29] and Godard et al.[30] proposed an image reconstruction using SfM techniques between stereo images to estimate the depth without a pose estimation.

Several studies were subsequently presented to improve accuracy. Casser et al. [11] introduced a motion mask based on semantic segmentation results to remove the effect of a dynamic object. Godard et al. [13] proposed monodepth2, which had a modified image reconstruction loss, considering occlusion. In addition, Kumar et al. [31] applied a recurrent neural network to understand the environment geometry from a video sequence. Pillai et al. [14] demonstrated that an input image with a higher resolution can enable a more accurate estimation, and Guizilini et al. [32] proposed an image reconstruction of different camera poses to efficiently update the neural network in semi-supervised learning. In addition to these emergent techniques, the following section presents most studies related to penalizing geometric inconsistencies between two point clouds estimated from different poses.

b) Penalization of Geometric Inconsistency: Mahjourian et al. [7] proposed a 3D point cloud alignment loss (iterative closest point (ICP) loss) to penalize the inconsistencies between two point clouds. Their approach employs an ICP to form the coupling between two point clouds estimated from the images captured at different poses. However, ICP is not differentiable. Hence, the ICP loss separately computes the pose inconsistencies and the residual error between two point clouds, with an approximation.

Gordon et al. [12] proposed a depth and ego-motion estimation system in the wild by learning the intrinsic camera parameters. In addition, they introduced a depth consistency loss [16] to minimize the difference between two estimated depth images from different frames. They shared a bi-linear sampler for the image reconstruction loss [5], [33] to determine the coupling and penalize any geometric inconsistencies. Because the coupling is determined from the estimated depth itself, the positive effect of their depth consistency loss can be limited because the estimated depth will be indefinite.



[a] baseline coupling

[b] our coupling

Fig. 3: **Overview of the baseline and our coupling.** The three red points in each image indicate the position of the estimated depth in the image space. The arrows between images at positions A and B indicate the coupling example. The number on the arrow is the assigned mass, which is delivered to the coupled points. The thickness of the arrow visually displays its assigned mass. Our coupling with the Wasserstein distance is a soft coupling with the preservation law, which can completely receive and deliver the same constant value (=$1/m$=$1/n$=1/3), and is symmetric.

Luo et al. [15] leveraged the optical flow to establish the coupling between point clouds and used these couplings to extract 3D geometric constraints. Because the optical flow is estimated using a pre-trained neural network, the coupling performance largely depends on the dataset domain. It has been reported that their approach does not work well on the KITTI dataset.

Fei et al. [9] introduced a semantically informed geometric loss to penalize deviations from a horizontal or vertical plane. They leveraged the results of semantic segmentation from a pre-trained network and an inertial measurement sensor to determine if the segmented areas belong to a horizontal or vertical plane. Although the surfaces of some objects can be an exact horizontal or vertical plane, e.g., a wall or road, their method cannot support most objects with complex shapes.

III. Proposed Method

A. Overview

Previous studies have shown that the penalization of a geometric inconsistency can enhance the depth and pose estimation accuracy [7], [12], [9], [15]. Inspired by these approaches, we propose a novel pluggable WCL, shown as L_{wass} in Fig. 2, for a depth and pose estimation. Here, L_{wass} works by adding it to the original objective L_{origin} from the target method. In addition, λ_w is a weighting factor of L_{wass} used to balance L_{origin}. In this paper, we focus on explaining L_{wass} instead of the mathematical formulation of L_{origin}, which is the same as in prior studies.

a) Preliminary: All prior approaches have commonly estimated the depth image as $D_A = f_{depth}(I_A)$, where $f_{depth}(\cdot)$ is a neural network for a depth estimation, and I_A is the image at camera pose A. The estimated depth image D_A can be projected onto a 3D point cloud as $Q_A^A[u,v] = D_A[u,v] \cdot K^{-1}[u,v,1]^T$. Here, u, v denotes the pixel position in the image coordinates, K denotes an intrinsic camera parameter, and Q_X^Y denotes a 3D point cloud at coordinate X from I_Y. It should be noted that K is a constant matrix, except for the baseline [12]. By calculating $Q_A^A[u,v]$ for the entire image space, we obtain the point cloud Q_A^A at coordinate A. By applying the same process to I_B, we can further obtain Q_B^B. In addition, Q_A^A and Q_B^B can be transformed into mutual coordinates as Q_B^A and Q_A^B by multiplying with the transformation matrices $T_{A\to B}$ and $T_{B\to A}$, respectively. Moreover, $T_{A\to B}$ and $T_{B\to A}$ are estimated from I_A and I_B, respectively, through another neural network, and (Q_A^A, Q_A^B) and (Q_B^B, Q_B^A) are sets of estimated point clouds at coordinates A and B, respectively.

b) Geometric consistency: Existing baseline objectives, which penalized geometric inconsistencies between Q_A^A and Q_A^B and between Q_B^B and Q_B^A, can be commonly represented as the following objective, L_{geo}:

$$L_{geo} = f(Q_A^A, Q_A^B) + f(Q_B^B, Q_B^A), \tag{1}$$

where $f(\cdot,\cdot)$ can be expressed as follows:

$$f(Q_X^X, Q_X^Y) = \sum_i^m \sum_j^n \mathrm{P}_{i,j} ||x_i - y_j||_2^2. \tag{2}$$

Here, x_i and y_j are the i-th and j-th estimated 3D points in Q_X^X and Q_X^Y, and m and n are the number of 3D points. In addition, $\mathrm{P}_{i,j}$ is a weighting value used to express the coupling between x_i and y_j. Ideally, we expect that $\mathrm{P}_{i,j}$ will be 1.0 if the corresponding 3D point to x_i in the real world is exactly the same as that of y_j; otherwise, $\mathrm{P}_{i,j}$ will be 0.0 in an ideal coupling.

To penalize the geometric inconsistency, the baselines shown in the related studies estimate $\mathrm{P}_{i,j}$, which is generally unknown. In an ICP loss [7], $\mathrm{P}_{i,j}$ is chosen as a binary value from ICP. And they approximately penalize L_{geo} for a depth estimation. In [12], $\mathrm{P}_{i,j}$ is given through a bi-linear sampling of the image reconstruction loss for a monocular depth estimation. In addition, Luo's method [15] leverages the optical flow between I_X and I_Y to estimate $\mathrm{P}_{i,j}$.

c) Benefits of our objective, WCL: To achieve a more suitable coupling between two point clouds, we propose WCL, which can have the following benefits against the baselines. The benefits of our WCL are as follows:

1) a geometrically consistent penalization,
2) smooth and symmetric objective, and a
3) simple implementation.

Fig. 3 shows a simple example of a coupling using three points($m = n = 3$). In the baseline coupling [7], [12], [15], multiple geometrically different points on one image can correspond to one point on the other image; that is, they violate the preservation law, where $\sum_i^m P_{i,j}$ and $\sum_j^n P_{i,j}$ are constant for all j and i, respectively. Further, the coupling of $A \to B$ and $B \to A$ can be asymmetric in those baselines, as shown in Fig. 3[a]. These behaviors are geometrically inconsistent. Note that we show $\frac{1}{3}(=\frac{1}{n}=\frac{1}{m})$ as the assigned $\mathrm{P}_{i,j}$ instead of 1.0 in Fig. 3[a], compared to our proposed WCL.

Furthermore, the baselines cause discontinuous coupling between each training iteration because updating a neural network for depth estimation changes the coupling, and the coupling itself is a hard coupling, where one point on I_A can make a coupling with only one point on I_B for $A \to B$ coupling. The same is true for $B \to A$ coupling, particularly in [7]. This means that a whole assigned value, "mass," is delivered to exactly one point. Here, "mass" is a term mainly used in an optimal transport [34].

Our proposed WCL can address these issues to improve the depth and pose estimation accuracy. In contrast to the baselines, our coupling is a soft coupling in which the sum of the delivered and received values is the same as the assigned mass $\frac{1}{3}$, as shown in Fig. 3[b]. Moreover, our soft coupling is symmetric, as shown in Fig. 3[b]. Hence, our WCL can enforce an avoidance of geometrically inconsistent penalization. In addition, the process of our WCL are completely differentiable without involving other external libraries (e.g., ICP) and result in a more stable learning compared to non-differentiable baselines [7], [9], [15]. As a result, the full algorithm can only be implemented on a GPU. Furthermore, the code size of our WCL is relatively short, as shown later.

B. WCL

We propose estimating $\mathrm{P}_{i,j}$ by minimizing $f(Q_X^X, Q_X^Y)$ in Eq. (2) under the *preservation law* discussed above, that is,

$$\min_{\mathrm{P}} \sum_{i=1}^m \sum_{j=1}^n \mathrm{P}_{i,j} \|x_i - y_j\|_2^2 \quad \underline{\text{subject to } \mathrm{P} \in \mathcal{U}_{m,n}}, \tag{3}$$

where $\mathcal{U}_{m,n}$ is a set of m-by-n matrices that satisfy the preservation law, which is formally defined by the following:

$$\mathcal{U}_{m,n} = \{\mathrm{P} \in \mathbb{R}_{\geq 0}^{m\times n} \mid \mathrm{P}\mathbf{1}_n = \frac{\mathbf{1}_m}{m}, \mathrm{P}^\top \mathbf{1}_m = \frac{\mathbf{1}_n}{n}\}, \tag{4}$$

where $\mathbf{1}_x$ is an x-dimensional vector, the elements of which are all 1s. Here, $\mathrm{P} \in \mathcal{U}_{m,n}$ can be a soft coupling, which can take a coupling with multiple points with the preservation law, maintaining $\mathrm{P}\mathbf{1}_n = \frac{\mathbf{1}_m}{m}$ and $\mathrm{P}^\top \mathbf{1}_m = \frac{\mathbf{1}_n}{n}$. With the optimal coupling P^* of the minimization problem in Eq. (3), we define $f(Q_X^X, Q_X^Y) = \sum_{i,j} \mathrm{P}_{i,j}^* \|x_i - y_j\|_2^2$ and rename L_{geo} as L_{wass}.

Example Before describing how to compute the optimal P^*, we elaborate on the definition above using an example in Fig. 4(a). Suppose two point clouds $Q_X^X = \{x_1, x_2, x_3\}$ and $Q_X^Y = \{y_1, y_2, y_3\}$, and consider a coupling (x_1, y_3), (x_2, y_1), and (x_3, y_2). This coupling clearly minimizes the total distance between coupled points; no other coupling can reduce the total distance (for example, the coupling in Fig. 4(c)). However, such an optimal coupling is unknown in advance; therefore, we need to determine the optimal

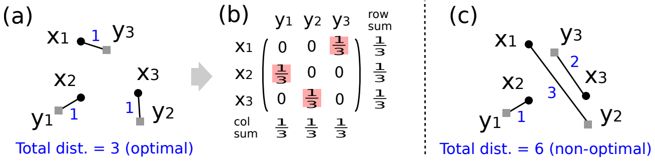


Fig. 4: **Coupling in WCL:** (a) optimal coupling, (b) matrix P corresponding to the coupling displayed in (a), and (c) non-optimal coupling; the blue numbers are the squared distances between two points. It should be noted that this figure is simplified. In fact, we can treat a soft coupling, which distributes the weight from one to many (see $\mathcal{U}_{m,n}$).

coupling that minimizes the total distance (for example, a non-optimal coupling in Fig. 4(c) does not provide a valid objective). The example in Fig. 4(b) represents the coupling matrix $\mathrm{P} \in \mathcal{U}_{m,n}$ corresponding to Fig. 4(a). The elements of P corresponding to the coupled points (i.e., (x_1, y_3), (x_2, y_1), and (x_3, y_2)) are filled with $\frac{1}{3}$, whereas the others are zeros. This P satisfies the constraint in Eq. (4), the sums of each row and column are $\frac{1}{m}$ and $\frac{1}{n}$, respectively (note that we have $m = n = 3$ here).

Importantly, the definition of our objective can be regarded as an instance in the (squared) *Wasserstein distance* [34], which was originally defined for probabilistic distributions.[1] It is known that the Wasserstein distance defines an appropriate metric, that is, it satisfies three metric axioms[2]; hence, we expect it to behave properly as an objective. Note that P provides soft coupling to allow weighted coupling with multiple points, although Fig.4 shows the simplified case with only one-to-one coupling. Further, Eq. (3) is well defined even if $m \neq n$, that is, the sizes of Q_X^X and Q_X^Y are different.

C. Computing WCL and its Gradient

To apply Eq. (3) as a loss function in end-to-end neural network training, we need an algorithm to compute the optimal $\mathrm{P}^*_{i,j}$, and the gradient with respect to the two input point clouds. For simplicity, we introduce an m-by-n matrix C as $\mathrm{C}_{i,j} = \|x_i - y_j\|_2^2$. We can rewrite the loss as $f(Q_X^X, Q_X^Y) = \langle \mathrm{P}^*, \mathrm{C} \rangle$ (where $\langle \cdot, \cdot \rangle$ is the inner product). To address the computational issues, we propose employing a *Sinkhorn iteration* (Algorithm 1), which allows us to compute $f(Q_X^X, Q_X^Y) = \langle \mathrm{P}^*, \mathrm{C} \rangle$ accurately along with its gradients [34]. The benefits of the Sinkhorn iteration are two-fold: i) *we can use a GPU* because it combines simple arithmetic operations; ii) *we can back-prop the iteration directly* through auto-gradient techniques equipped in most modern deep learning libraries (i.e., we do not need external libraries, unlike the ICP loss).

Algorithm 1: Computing WCL $f(Q_X^X, Q_X^Y)$ using the Sinkhorn iteration. Here, $\varepsilon > 0$ is a small constant.

```
  Data: Point clouds $Q_X^X = \{x_i\}_{i=1}^m$, $Q_X^X = \{y_j\}_{j=1}^n$
1 $\mathrm{C}_{ij} = \|x_i - y_j\|_2^2 \quad 1 \leq \forall i \leq m,\ 1 \leq \forall j \leq n$
2 $\mathrm{G} \leftarrow \exp(-\mathrm{C}/\varepsilon)$            // Element-wise exp
3 $\mathrm{v} \leftarrow \mathbf{1}_n$                // Initialize dual variable
4 while Not converged do
5     $\mathrm{u} \leftarrow \frac{1}{m}\mathbf{1}_m/\mathrm{G}\mathrm{v}$          // Element-wise division
6     $\mathrm{v} \leftarrow \frac{1}{n}\mathbf{1}_n/\mathrm{G}^\top\mathrm{u}$
7 return $\langle \mathrm{diag}(\mathrm{u})\mathrm{G}\,\mathrm{diag}(\mathrm{v}), \mathrm{C} \rangle$ as $f(Q_X^X, Q_X^Y)$
```

Remark. To be exact, Algorithm 1 is an approximation algorithm for Eq. (3), which solves the optimization problem with an additional *entropy regularization* term $H(\mathrm{P})$:

$$\min_{\mathrm{P}} \langle \mathrm{P}, \mathrm{C} \rangle - \varepsilon H(\mathrm{P}), \quad \text{subject to } \mathrm{P} \in \mathcal{U}_{m,n} \tag{5}$$

$$\text{where} \quad H(\mathrm{P}) := -\sum_{i,j} \mathrm{P}_{i,j}(\log \mathrm{P}_{i,j} - 1).$$

Using the optimal solution $\mathrm{P}^\dagger$ to this problem, we define the *regularized* WCL by $f(Q_X^X, Q_X^Y) = \langle \mathrm{P}^\dagger, \mathrm{C} \rangle$. This regularization makes the WCL smooth with respect to its inputs, resulting in stable training. If $\varepsilon \to 0$, Eq. (5) converges to the original optimization problem in Eq. (3). Hence, a small $\varepsilon > 0$ provides a good approximation. However, such a small ε might cause numerical instability because a small ε reduces G to an almost zero matrix on line 2. To improve the numerical stability, we may use log-sum-exp on lines 5 and 6. Further, we can compute Algorithm 1 in parallel over the batch dimensions. On line 4, we can use any type of stopping condition; here, we stop at n_{it} iterations. See [35], [34] for details of the entropy regularization.

IV. EXPERIMENT

We applied our WCL to prior studies and quantitatively and qualitatively evaluated them in comparison to the original method using a public dataset.

A. Dataset

In this study, we used the KITTI dataset [38]. Although it is known that a large input image size can lead to a better performance [14], the main purpose of our experiment is to confirm advantages of the proposed method over the original method applied. Hence, we use an image size of 416×128, similar to previous studies [5], [7], [12], [13]. However, we conducted an additional evaluation using a pixel resolution of 640×192 for monodepth2 only [13] because this is the default image size in the public code. As with several prior studies, we have two different dataset splits for the evaluation of the depth and pose estimation. It should be noted that the models for depth and pose estimation are trained simultaneously on each dataset, whereas the depth and pose are trained and evaluated on separate data.

a) Depth Estimation: We separate the KITTI raw dataset through an Eigen split [39]. As a result, we have approximately 40,000 frames for training, 4,000 frames for validation, and 697 frames for testing. The test data are

[1] Point clouds can currently be regarded as mixtures of delta functions; hence, we can deal with point clouds using the Wasserstein distance.

[2] That is, $f(Q_X^X, Q_X^Y)$ satisfies the following: (i) $f(Q_X^X, Q_X^Y) = 0 \Leftrightarrow Q_X^X = Q_X^Y$ (identity of indiscernible aspects); (ii) $f(Q_X^X, Q_X^Y) = f(Q_X^Y, Q_X^X)$ (symmetry); (iii) $f(Q_X^X, Q_X^Y) \leq f(Q_X^X, Q_X^Z) + f(Q_X^Z, Q_X^Y)$ (triangle inequality).

TABLE I: **Evaluation of the depth estimation through a self-supervised mono supervision on an Eigen split of the KITTI dataset.** We display seven metrics from the **estimated depth images of less than 80 m**. Here, "+ WCL" is the result obtained with our proposed method and the baseline method presented above. For the four leftmost metrics, a smaller value is better; for the three rightmost metrics, a higher value is better. In addition, † and ‡ indicate the removal of the ICP and depth consistency loss from the original cost function, respectively. The method of ‡ was evaluated by the author. The $*$ method uses the ground truth pose.

Method	image size	Abs-Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Yang et al. [36]	416x128	0.182	1.481	6.501	0.267	0.725	0.906	0.963
LEGO [6]	416x128	0.162	1.352	6.276	0.252	0.783	0.921	0.969
GeoNet [37]	416x128	0.155	1.296	5.857	0.233	0.793	0.931	0.973
Fei et al. [9]	416x128	0.142	1.124	5.611	0.223	0.813	0.938	0.975
DDVO [8]	416x128	0.151	1.257	5.583	0.228	0.810	0.936	0.974
Yang et al. [10]	416x128	0.131	1.254	6.117	0.220	0.826	0.931	0.973
Casser et al. [11]	416x128	0.141	1.026	5.291	0.2153	0.8160	0.9452	0.9791
Luo et al. [15] $*$	384x112	0.130	2.086	4.876	0.205	0.878	0.946	0.970
Zhou et al. [5]	416x128	0.208	1.768	6.856	0.283	0.678	0.885	0.957
+ WCL ($n_c = 16$, $n_r = 4$)	416x128	**0.171**	**1.316**	**6.080**	**0.255**	**0.755**	**0.915**	**0.966**
vid2depth [7]	416x128	**0.163**	1.240	6.220	0.250	0.762	0.916	**0.968**
vid2depth [7] wo ICP loss†	416x128	0.175	1.617	6.267	0.252	0.759	0.917	0.967
+ WCL ($n_c = 16$, $n_r = 4$)	416x128	0.165	**1.226**	**5.892**	**0.246**	**0.767**	**0.918**	**0.968**
Gordon et al. [12]	416x128	0.128	0.959	5.23	0.212	**0.845**	**0.947**	0.976
Gordon et al. [12] wo depth consis. loss‡	416x128	0.129	0.945	**5.211**	0.214	0.839	0.944	0.976
+ WCL ($n_c = 16$, $n_r = 4$)	416x128	**0.125**	**0.915**	5.231	**0.210**	0.844	**0.947**	**0.977**
monodepth2 [13]	416x128	0.128	1.087	5.171	0.204	0.855	**0.953**	0.978
+ WCL ($n_c = 32$, $n_r = 8$)	416x128	0.126	0.933	5.039	0.203	0.853	0.952	0.979
+ WCL ($n_c = 32$, $n_r = 4$)	416x128	0.125	0.938	5.006	0.202	0.854	**0.953**	**0.980**
+ WCL ($n_c = 16$, $n_r = 8$)	416x128	0.125	0.933	5.039	0.203	0.853	0.952	0.979
+ WCL ($n_c = 16$, $n_r = 4$)	416x128	**0.123**	**0.920**	**4.990**	**0.201**	**0.858**	**0.953**	**0.980**
monodepth2 [13]	640x192	0.115	0.903	4.863	0.193	**0.877**	**0.959**	**0.981**
+ WCL ($n_c = 16$, $n_r = 4$)	640x192	**0.114**	**0.813**	**4.705**	**0.191**	0.874	**0.959**	**0.981**

chosen from 29 scenes of the KITTI raw dataset. Although the KITTI raw data include stereo images and LiDAR data, we use monocular images for both training and testing as the input data and use the LiDAR data as the ground truth for testing only.

b) Pose Estimation: Unlike the KITTI raw dataset, the KITTI odometry dataset has the ground truth of the relative pose between consecutive images for testing. However, test data in the KITTI odometry dataset are partially included in the training data of the KITTI raw dataset. Hence, we cannot use the trained model on the KITTI raw dataset for the pose estimation evaluation on the KITTI odometry dataset. Therefore, we apply both training and testing on the KITTI odometry dataset, similar to the baseline methods [5], [13]. Although there are 22 sequences in the KITTI odometry dataset, we used 11 sequences, from 00 to 10. After training the models on sequences 00 through 08, we tested the model for pose estimation on sequences 09 and 10.

B. Evaluation of Depth Estimation

a) Training: In the evaluation of the depth estimation, we apply our WCL to four selected baselines [5], [7], [12], [13]. For [5], [13], we add $\lambda_w \cdot L_{wass}$ to their original cost function, L_{origin}, to train the model. By contrast, because the original cost function of [7], [12] includes the ICP loss [7] and the depth consistency loss [12], which also penalizes the geometric inconsistencies, we remove them to apply our WCL for the comparative evaluation with each objective. We train all models by applying the same hyperparameters (e.g., mini-batch size, learning rate, data augmentation, and network structure) and training process (for example, masking, training length, optimization, and selection of input images I_A and I_B in Fig. 2) as in the original code, except for the following hyperparameters of our WCL.

We determine the weighting values, λ_w, for L_{wass} as 7.0, 2.0, 3.0, and 0.5 in [5], [7], [12], and [13], respectively. Here, λ_w is a weighting factor that balances L_{origin}; therefore, there is no significance to the relative size of the values. In addition, ε in Eq. (5) is set to 0.001 to suppress the approximation and stably calculate the WCL. We set the number of Sinkhorn iterations n_{it} to 30 to obtain a better coupling under the GPU memory limitation. In addition, we uniformly sample the point clouds at a grid point on an image coordinate before feeding them into our WCL in Eq. (1) owing to the GPU memory limitation. To allow more point clouds in our WCL to be applied, the vertical and horizontal grid point intervals are determined as $n_c = 16$ and $n_r = 4$. During training, to cover all point clouds, we have random offsets for the grid point, which are less than or equal to n_c and n_r on the vertical and horizontal axes. The ablation study of n_c and n_r is shown in the next subsection.

b) Quantitative Analysis: Table I displays the seven widely used metrics for an evaluation of the depth estimation from monocular camera images. As summarized in Table I, the performance of all baselines can be successfully improved on most metrics by adding our WCL. The improvements for [5] and [13] are approximately 25% and 15% on the most advantageous metric, respectively. However, it is not as large for [7] and [12] because they also penalize the geometric inconsistencies using their own approach. However, our method still has explicit margins of approximately

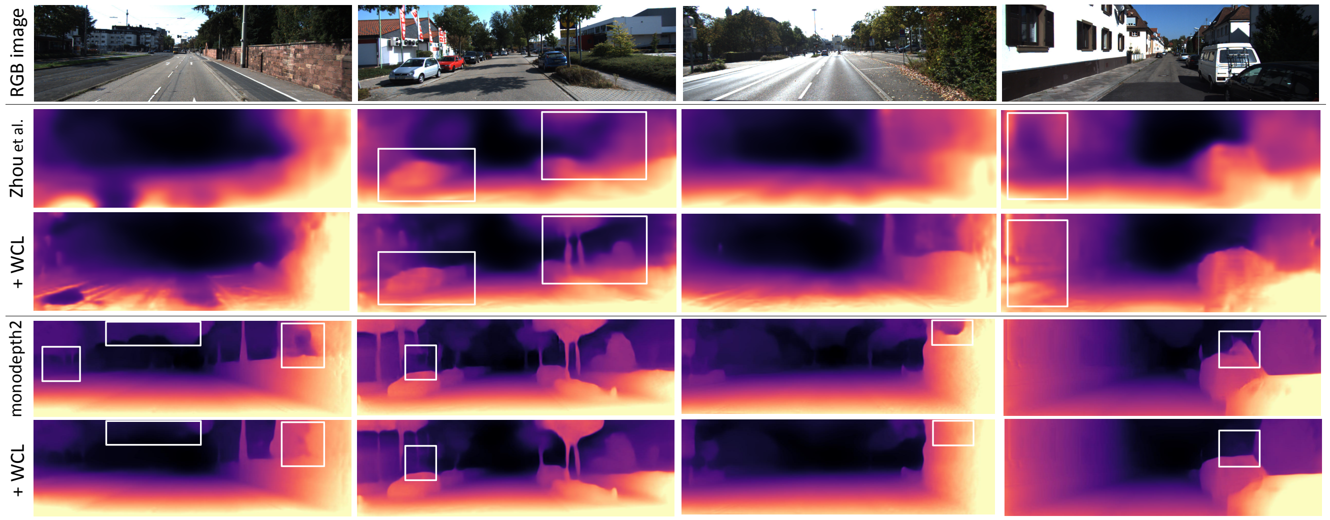


Fig. 5: **Qualitative results of our proposed method.** The top row displays the input images for the trained neural network to estimate depth images, and the other rows display the estimated depth images, with and without the WCL. The white rectangular box in the depth image highlights the advantages of the WCL.

5% on Sq Rel or RMSE against these baselines. Furthermore, "monodepth2 [13] + WCL" can successfully obtain the best results, which are better than those of the other related studies [15], [9].

In addition, we evaluated n_c and n_r in monodepth2. A better performance can be obtained at a smaller n_c and n_r because geometric inconsistencies can be accurately penalized. However, because it is difficult to make n_c and n_r even smaller owing to the GPU memory limitations, we commonly set $n_c = 16$ and $n_r = 4$ for our method.

c) Qualitative Analysis: We show the depth images of [5], [13], with and without our WCL, in Fig. 5. The first row in Fig. 5 shows the RGB image as the neural network input, the second and fourth rows are the depth images estimated by [5] and [13], and the third and fifth rows are the depth images estimated by "[5] + WCL" and "[13] + WCL", respectively. In the estimated depth images, we display a white lined rectangle to highlight the advantage of our method. Our method can reduce the number of artifacts and sharpen the estimation.

C. Evaluation of Pose Estimation

For an evaluation of the pose estimation, we select two prior studies [5], [13] to apply our WCL. We were unable to evaluate [7], [12] for a pose estimation because the hyper-parameters and codes for the pose estimation and evaluation are partially unclear from their released code. For [5], [13], we trained the models using the KITTI odometry dataset under the same training conditions as the depth estimation.

Table II shows the absolute trajectory error (ATE) in meters for sequences 09 and 10 of the KITTI odometry dataset. We show the mean and standard deviation of the ATE over all five overlapping frame snippets. By applying our WCL, the pose estimation accuracies of two prior studies [5], [13] are improved. Although the performance is slightly worse than that of "ORB-SLAM (full)," which is the original ORB-SLAM with a loop closure using whole sequence images, the methods applying our WCL outperform "ORB-SLAM (short)," which takes five consecutive images as with our approach.

TABLE II: **Evaluation of pose estimation on sequences 09 and 10 of the KITTI odometry dataset.** The results show the mean and standard deviation of the absolute trajectory error in meters.

Method	image size	sequence 09	sequence 10
ORB-SLAM (full)	full size	0.014 ± 0.008	0.012 ± 0.011
ORB-SLAM (short)	full size	0.064 ± 0.141	0.064 ± 0.130
Zhou et al. [5]	416x128	0.021 ± 0.017	0.020 ± 0.015
+ WCL	416x128	**0.016** ± **0.011**	**0.013** ± **0.009**
monodepth2 [13]	416x128	0.017 ± **0.009**	0.015 ± **0.010**
+ WCL	416x128	**0.016** ± **0.009**	**0.014** ± **0.010**
monodepth2 [13]	640x192	0.017 ± **0.008**	0.015 ± **0.010**
+ WCL	640x192	**0.016** ± **0.008**	**0.014** ± **0.010**

V. CONCLUSION

In this paper, we proposed a novel WCL to penalize the geometric inconsistencies for a depth and pose estimation. Our proposed approach employs the Wasserstein distance to measure the consistency between two point clouds from different frames. Our WCL is a smooth and symmetric objective, which can suitably measure the geometric consistencies without using any other external and/or non-differentiable libraries. Therefore, the neural network can be effectively and efficiently trained to obtain highly accurate depth estimations. During the experiment, we applied our proposed WCL to several state-of-the-art baseline approaches and confirmed the advantages of our method with healthy margins.

As shown in the experiments, the performance of our WCL is restricted by the limitation on the GPU memory. In addition, our WCL has potentials to improve the performance by considering occlusions and dynamic objects like the image reconstruction loss [10], [11], [13], [12]. Addressing these issues will be an important future direction.

References

[1] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[2] J. Biswas and M. Veloso, "Depth camera based indoor mobile robot localization and navigation," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1697–1702.

[3] N. Hirose, F. Xia, R. Martín-Martín, A. Sadeghian, and S. Savarese, "Deep visual mpc-policy learning for navigation," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3184–3191, 2019.

[4] N. Hirose, A. Sadeghian, M. Vázquez, P. Goebel, and S. Savarese, "Gonet: A semi-supervised deep learning approach for traversability estimation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3044–3051.

[5] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858.

[6] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, "Lego: Learning edge with geometry all at once by watching videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 225–234.

[7] R. Mahjourian, M. Wicke, and A. Angelova, "Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5667–5675.

[8] C. Wang, J. Miguel Buenaposada, R. Zhu, and S. Lucey, "Learning depth from monocular videos using direct methods," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2022–2030.

[9] X. Fei, A. Wong, and S. Soatto, "Geo-supervised visual depth prediction," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1661–1668, 2019.

[10] Z. Yang, P. Wang, Y. Wang, W. Xu, and R. Nevatia, "Every pixel counts: Unsupervised geometry learning with holistic 3d motion understanding," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.

[11] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8001–8008.

[12] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, "Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8977–8986.

[13] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, "Digging into self-supervised monocular depth estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3828–3838.

[14] S. Pillai, R. Ambruş, and A. Gaidon, "Superdepth: Self-supervised, super-resolved monocular depth estimation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9250–9256.

[15] X. Luo, H. Jia-Bin, R. Szeliski, K. Matzen, and J. Kopf, "Consistent video depth estimation," *arXiv preprint arXiv:2004.15021*, 2020.

[16] J.-W. Bian, Z. Li, N. Wang, H. Zhan, C. Shen, M.-M. Cheng, and I. Reid, "Unsupervised scale-consistent depth and ego-motion learning from monocular video," *arXiv preprint arXiv:1908.10553*, 2019.

[17] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 2017, pp. 214–223.

[18] G. Huang, C. Guo, M. J. Kusner, Y. Sun, F. Sha, and K. Q. Weinberger, "Supervised word mover's distance," in *Advances in Neural Information Processing Systems*, 2016, pp. 4862–4870.

[19] N. Courty, R. Flamary, and M. Ducoffe, "Learning wasserstein embeddings," in *International Conference on Learning Representations*, 2018.

[20] N. Courty, R. Flamary, D. Tuia, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 9, pp. 1853–1865, 2017.

[21] B. Ummenhofer, H. Zhou, J. Uhrig, N. Mayer, E. Ilg, A. Dosovitskiy, and T. Brox, "Demon: Depth and motion network for learning monocular stereo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5038–5047.

[22] V. Patil, W. Van Gansbeke, D. Dai, and L. Van Gool, "Don't forget the past: Recurrent depth estimation from monocular video," *arXiv preprint arXiv:2001.02613*, 2020.

[23] A. Johnston and G. Carneiro, "Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4756–4765.

[24] Z. Zhang, S. Lathuiliere, E. Ricci, N. Sebe, Y. Yan, and J. Yang, "Online depth learning against forgetting in monocular videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4494–4503.

[25] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3d packing for self-supervised monocular depth estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[26] L. Tiwari, P. Ji, Q.-H. Tran, B. Zhuang, S. Anand, and M. Chandraker, "Pseudo rgb-d for self-improving monocular slam and depth prediction," *arXiv preprint arXiv:2004.10681*, 2020.

[27] V. Guizilini, R. Hou, J. Li, R. Ambrus, and A. Gaidon, "Semantically-guided representation learning for self-supervised monocular depth," in *International Conference on Learning Representations (ICLR)*.

[28] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki, "Sfm-net: Learning of structure and motion from video," *arXiv preprint arXiv:1704.07804*, 2017.

[29] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *European conference on computer vision*. Springer, 2016, pp. 740–756.

[30] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 270–279.

[31] A. CS Kumar, S. M. Bhandarkar, and M. Prasad, "Depthnet: A recurrent neural network architecture for monocular depth prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 283–291.

[32] V. Guizilini, J. Li, R. Ambrus, S. Pillai, and A. Gaidon, "Robust semi-supervised monocular depth estimation with reprojected distances," in *Conference on Robot Learning*, 2020, pp. 503–512.

[33] M. Jaderberg, K. Simonyan, A. Zisserman, *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[34] G. Peyré, M. Cuturi, *et al.*, "Computational optimal transport," *Foundations and Trends® in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[35] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," in *Advances in neural information processing systems*, 2013, pp. 2292–2300.

[36] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia, "Unsupervised learning of geometry with edge-aware depth-normal consistency," *arXiv preprint arXiv:1711.03665*, 2017.

[37] Z. Yin and J. Shi, "Geonet: Unsupervised learning of dense depth, optical flow and camera pose," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1983–1992.

[38] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[39] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," in *Advances in neural information processing systems*, 2014, pp. 2366–2374.