We propose a novel objective to penalize geometric inconsistencies, to improve the performance of depth estimation from monocular camera images. Our objective is designed with 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 keeps geometric constraints and leads differentiable objective. By adding our objective to the original ones of other state-of-the-art methods, we can effectively penalize a geometric inconsistency and obtain a highly accurate depth estimation. Our proposed method is evaluated on the Eigen split of the KITTI raw dataset.

**Keywords:** Monocular Depth Estimation, Wasserstein Distance, Geometry

1 Introduction

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

Self-supervised learning using videos captured by robots and autonomous vehicles enables the learning of depth estimation networks without ground truth depth images. The image reconstruction loss proposed by Zhou et al. [3] has significantly improved the depth estimation accuracy, and various pluggable objectives, network structures, data augmentation, and masking methods for dynamic and occluded objects have been suggested [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. However, there are still limitations for environments and objects far from the camera posture and having a uniform color.

In this paper, we propose a novel additional objective to evaluate a geometric consistency, to improve the performance of depth estimation. The proposed objective uses the Wasserstein distance to measure the consistency between two point clouds estimated from images at different poses.

Several previous studies have proposed objectives that attempt to penalize an inconsistency on 3D geometric constraints [6, 9, 11, 13]. However, their performance is limited owing to a strong approximation for the evaluation of consistency [6, 11] and the dependence on other incomplete algorithms to obtain the correspondences between point clouds [6, 13]. In contrast, our method attempts to directly measure a geometric consistency from 3D point clouds without any indirect process and bold approximation. In addition, the mathematical formulation of our objective is smooth and symmetric, which is advantageous for an efficient and effective training process. Thus, we aim to enhance the accuracy of depth estimation compared to that achieved using the conventional methods. The major contributions of this study are,

1. proposal of a novel objective, the Wasserstein consistency loss (WCL), to evaluate consistency on 3D geometric constraints,
2 Related Work

Self-supervised depth estimation has become popularized recently [3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. In this section, we introduce most related studies that attempt to penalize an inconsistency based on 3D geometric constraints.

Mahjourian et al. [6] 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. Because an ICP is not differentiable, the ICP loss separately computes the rotation and translation inconsistencies and the residual error between two point clouds after the ICP alignment.

Gordon et al. [9] mainly proposed the depth and ego-motion estimation by learning the camera intrinsic parameters. In addition, they introduced a depth consistency loss to minimize the difference between two estimated depth images from different frames. They shared a bi-linear sampler for the image reconstruction loss [3, 14], to make the coupling between both the estimated depth images and penalize a geometric inconsistency.

Luo et al. [13] fine-tuned a pretrained network for depth estimation to minimize the spatial and disparity losses extracted from 3D geometric constraints. They leveraged the optical flow to establish the coupling between point clouds, and used these coupling for extracting 3D geometric constraints.

Fei et al. [11] introduced a semantically informed geometric loss to penalize the deviation from a horizontal or vertical plane at specific segmented areas. They leveraged the results of the semantic segmentation from a pre-trained network and an inertial measurement sensor, to determine if the segmented areas belonged to a horizontal or a vertical plane.

3 Proposed Method

3.1 Overview

From previous studies, it is known that the penalization of a geometric inconsistency can enhance the accuracy of depth estimation [6, 9, 11, 13]. In comparison to other approaches, we propose a novel additional objective, WCL, for depth estimation from monocular camera images. Fig. 2 displays the overview of our proposed approach. Here, \(I_X\) is the image at camera pose \(X\). \(Q_X\) denotes the point cloud estimated from \(I_X\) on coordinate \(Y\). The origin of coordinate \(X\) is equal to camera pose \(X\). Similar to [6, 9, 13], our WCL penalizes the inconsistency between two point clouds having the
Figure 2: Overview of our proposed approach. We feed RGB images $I_A$ and $I_B$ into a neural network to estimate depth images $D_A$ and $D_B$. From $D_A$ and $D_B$, we obtain point clouds $Q_{A_A}$, $Q_{B_B}$, $Q_{A_B}$, and $Q_{B_A}$ using the camera intrinsic parameters and the estimated transformation matrices, $T_{A \rightarrow B}$ or $T_{B \rightarrow A}$. To penalize a geometric inconsistency, we propose the addition of the WCL $L_{\text{wass}}$ to the original cost function, $L_{\text{origin}}$, of recent state-of-the-art approaches, to train the neural networks.

Figure 3: Overview of baseline and our coupling. Two red points in each image indicate the position of estimated depth on image space. Arrows between $t$-th and $t + 1$-th images indicate the coupling. The number on the arrow is the assigned weighting value, which is delivered to the coupled points. The thickness of the arrow visually displays its delivered value. Our coupling with the Wasserstein distance is soft coupling with the preservation law, which can totally receive and deliver same constant value ($\approx 0.5$). Different from the baseline, our correspondence is symmetric.

The same geometry. The WCL is defined as $L_{\text{wass}}$:

$$L_{\text{wass}} = \tilde{W}^2(Q_{A_A}, Q_{B_B}) + \tilde{W}^2(Q_{B_B}, Q_{A_A}),$$

(1)

Here, $\tilde{W}^2(\cdot)$ is the Wasserstein distance between two point clouds on the same coordinate, as displayed subsequently. We add $\lambda_w \cdot L_{\text{wass}}$ to the original objective, $L_{\text{origin}}$, of the recent state-of-the-art methods and update the neural network by minimizing the combined objective for penalizing a geometric inconsistency. $\lambda_w$ is a weighting factor of $L_{\text{wass}}$ to balance $L_{\text{origin}}$. The benefits of our WCL are,

1. intuitive penalization of a 3D geometric inconsistency,
2. smooth and symmetric objective,
3. simple implementation.

Our WCL employs the Wasserstein distance to intuitively calculate a 3D geometric consistency using 3D point clouds. The evaluation of the geometric consistency is calculated in two steps: 1) taking a coupling between two point clouds and 2) calculation of the distance (e.g. Euclidean distance) between the aligned point clouds using the coupling. Fig. 3 shows the overview of both baseline and our coupling with the Wasserstein distance on the image space. Our coupling is a soft coupling in which the mass assigned on each point is preserved (Fig. 3[b]), sum of delivered and received values are same as the assigned mass (0.5 in Fig. 3). In Fig. 3[b], there is no point that does not take a coupling as in Fig. 3[a] [6, 9, 11, 13]. It means that our soft coupling in Fig. 3[b] can avoid geometrically wrong training, in which one point will be geometrically same as two different points, in this case. This conservation law can be said to be in accordance with geometric constraints. In addition, since the geometric positions of the estimated point clouds from different two frames do not always match, our soft coupling is more suitable.

Moreover, our soft coupling by the Wasserstein distance is symmetric, as shown in Fig. 3[b], whereas the baselines have different coupling depending on the reference point cloud, as shown in Fig. 3[a].
Furthermore, the baselines cause discontinuous coupling between training iterations, because updating of depth estimation switches the coupling. Hence, the cost function itself becomes discontinuous, worsening the convergence of learning. In contrast, the proposed method takes the soft coupling, leading to a differentiable cost function without involving other external libraries and more stable learning than the non-differentiable baselines [6, 11, 13]. As the result, this entire algorithm can be implemented only on a GPU. And, the code size of our WCL is relatively short as shown later.

3.2 WCL

We introduce the Wasserstein distance [15] and its approximation \( \hat{W}^2(\cdot) \) used in \( L_{wass} \), which provides a differentiable loss function that penalizes the geometric inconsistency between two point clouds.

**Preliminary: Definition and Notation** A point cloud is a set of points in \( \mathbb{R}^3 \). Let \( X = \{x_1, \cdots, x_m\} \), where \( x_i \in \mathbb{R}^3 \), and \( X \) is a point cloud of size \( m \). \( \langle U, V \rangle \) denote the inner product between two vectors (or matrices) of the same size. \( \mathbb{I}_n \) is an \( n \)-dimensional vector all whose elements are one.

**Wasserstein Distance** Let \( X \) and \( Y \) be point clouds of size \( m \) and \( n \), respectively. The Wasserstein distance\(^2\) is a metric between two point clouds, e.g., \( X \) and \( Y \). This is defined as the sum of the squared Euclidean distances between two “coupled” points in \( X \) and \( Y \). An example is shown in Fig. 4(a). Suppose two point clouds \( X = \{x_1, x_2, x_3\} \) and \( Y = \{y_1, y_2, y_3\} \), and consider a coupling \( (x_1, y_3), (x_2, y_1), (x_3, y_2) \). This coupling clearly minimizes the total distance between coupled points; no other coupling can reduce the total distance (e.g., the coupling in Fig. 4(c)). However, such an optimal coupling is unknown in advance; therefore, we must find the optimal coupling, which minimizes the total distance (e.g., a non-optimal coupling in Fig. 4(c) does not provide the valid Wasserstein distance). Essentially, identifying such an optimal matching corresponds to the following optimization problems [15]:

\[
\min_{P} \langle P, C \rangle \quad \text{subject to } P \in U_{m,n}. \tag{2}
\]

Here, \( C \) is an \( m \)-by-\( n \) Euclidean distance matrix whose \((i, j)\) element is the squared distance between \( x_i \) and \( y_j \) (i.e., \( C_{ij} = \|x_i - y_j\|^2 \)). \( P \) is an \( m \)-by-\( n \) coupling matrix whose \((i, j)\) element has a matching weight between \( x_i \) and \( y_j \). Moreover, \( U_{m,n} \) is a set of \( m \)-by-\( n \) matrices that represent valid coupling, which is formally defined by

\[
U_{m,n} = \{ P \in \mathbb{R}^{m \times n}_{\geq 0} \mid \mathbb{P} \mathbb{I}_n = \mathbb{I}_m, P^\top \mathbb{I}_m = \mathbb{I}_n \}. \tag{3}
\]

This suggests that the mass assigned on each point in \( X \) (i.e., \( \frac{1}{m} \)) must be delivered to points in \( Y \) without overs and shorts, and vice versa; this conservation law is entirely different from the existing geometric inconsistency losses. The example in Fig. 4(b) represents the coupling matrix corresponding to Fig. 4(a). Elements of \( P \) corresponding to the coupled points (i.e., \((x_1, y_3), (x_2, y_1), (x_3, y_2) \)) are filled with \( \frac{1}{3} \), while the others are zero. This \( P \) satisfies the constraint (3); the sums of each row and column are \( \frac{1}{3} \) and \( \frac{1}{3} \) (note that we have \( m = n = 3 \) here).

With the optimal solution \( P^* \), the Wasserstein distance is defined by \( W^2(X, Y) = \langle P^*, C \rangle \). It is known that \( W^2(\cdot, \cdot) \) defines a proper metric, i.e., the Wasserstein distance satisfies three axioms\(^3\) of metric [15]; hence, we expect that it behaves properly as a loss function. Note that \( P \) provides soft coupling to allow weighted coupling with multiple points; however, Fig. 4 shows the simplified case only with one-to-one matching; hence, we can define (2) even if \( m \neq n \), i.e., size of \( X \) and \( Y \) are different.

---

\(^2\) Originally, the Wasserstein distance was defined for probabilistic distributions. Currently, point clouds can be regarded as mixtures of delta functions; thus, we can deal with point clouds with the Wasserstein distance.

\(^3\) That is, \( W(X, Y) \) satisfies: (i) \( W(X, Y) = 0 \Leftrightarrow X = Y \) (identity of indiscernibles); (ii) \( W(X, Y) = W(Y, X) \) (symmetry); (iii) \( W(X, Y) \leq W(X, Z) + W(Z, Y) \) (triangle inequality).
Figure 4: Wasserstein distance. (a) Optimal coupling; (b) Matrix P corresponding to coupling displayed in (a); (c) Non-optimal coupling; Blue numbers are squared distances between two points. Note that this figure is simplified; we can in fact treat soft coupling, which distributes the weight from one to many (see $U_{m,n}$).

Computing WCL and its Gradient As described in Eq. (1), we train neural networks by reducing the geometric inconsistency of two point clouds measured by Wasserstein distance. To this end, we need to compute $W^2(X,Y)$ and its gradients w.r.t. $X$ and $Y$. Sinkhorn iteration (Algorithm 1) allows us to compute $W^2(X,Y) = \langle P^*, C \rangle$ very accurately, as well as its gradients. Benefits of the Sinkhorn iteration are two-fold: (1) we can use GPUs because it is a combination of simple arithmetic operations; (2) we can backprop the iteration directly through auto-gradient techniques equipped in most of modern deep learning libraries (i.e., we do not need external libraries, unlike the ICP loss).

Algorithm 1: Computing WCL $\widetilde{W}^2(X,Y)$ by the Sinkhorn iteration. $\varepsilon > 0$ is a small constant.

**Data:** Point clouds $X = \{x_i\}_{i=1}^m, Y = \{y_j\}_{j=1}^n$

1. $C_{ij} = \|x_i - y_j\|^2, 1 \leq i \leq m, 1 \leq j \leq n$
2. $K \leftarrow \exp(-C/\varepsilon)$ // Element-wise exp
3. $v \leftarrow \mathbb{1}_n$ // Initialize dual variable
4. **while** Not converged **do**
5. $u \leftarrow \frac{1}{m} \mathbb{1}_m / K v$ // Element-wise division
6. $v \leftarrow \frac{1}{n} \mathbb{1}_n / K^\top u$
7. **return** $\langle \text{diag}(u) K \text{diag}(v), C \rangle$ as $\widetilde{W}^2(X,Y)$

Remark To be exact, Algorithm 1 is an approximation algorithm for (2); it solves the optimization problem with an additional entropy regularization term $H(P)$:

$$\min_P \langle P, C \rangle - \varepsilon H(P), \text{ subject to } P \in U_{m,n} \quad (4)$$

where $H(P) := -\sum_{ij} P_{ij} (\log P_{ij} - 1)$.

With the optimal solution $P^\dagger$ of this problem, we define the regularized Wasserstein distance by $\widetilde{W}^2(X,Y) = \langle P^\dagger, C \rangle$. This regularization term makes WCL smooth w.r.t. its inputs, which leads to a stable training. As we see, if $\varepsilon \rightarrow 0$, (4) converges to the original optimization problem (2); hence, a small $\varepsilon > 0$ gives a good approximation; however such a small $\varepsilon$ might cause numerical instability, because a small $\varepsilon$ makes $K$ almost a zero matrix at Line 2. To improve numerical stability, we may use log-sum-exp at Lines 5–6. Further, we can compute Algorithm 1 in parallel over batch dimension. At Line 4, we can use any kind of stopping condition; we simply iterate 100 times in this paper. See [16, 15] for details of the regularized Wasserstein distance.

4 Experiment

We apply our WCL into four prior works [3, 6, 9, 10] and quantitatively and qualitatively evaluate them on public dataset.
Figure 5: Sparse sampling of point clouds. Red dots are position of sampled point clouds on image space.

4.1 Dataset

We commonly use the KITTI raw dataset [17], in this experiment. The image size is taken as $416 \times 128$, in our evaluation. As the result of the Eigen split [18], we have approximately 40,000 frames for training, 4,000 frames for validation, and 697 frames for the test. Test data is chosen from 29 scenes of the KITTI raw dataset. Although the KITTI raw data includes stereo images and LIDAR data, we only use monocular images for both training and testing as the input data and use the LIDAR data as only the ground truth for testing.

4.2 Sparse Sampling

We uniformly sample the point clouds to reduce the total number of the points only for the calculation of WCL, because of limitation of memory usage on GPU. Fig. 5 shows the sampled points on the depth image space. Here, $n_c$ and $n_r$ indicate vertical and horizontal interval of grid points. $m_c$ and $m_r$ indicate random offset less than and equal to $n_c$ and $n_r$ to cover the whole point clouds in training.

4.3 Training

We train the models with $L_{wass}$, to verify the effectiveness of our method on the KITTI dataset. In this experiment, we select four baselines [3, 6, 9, 10] and add $\lambda_w \cdot L_{wass}$ to their original cost function, $L_{origin}$, to train the model. Note that we remove the ICP loss of [6] and the depth consistency loss of [9], which also penalize a geometric inconsistency from their $L_{origin}$ when the WCL is added. In the training, we apply the same hyperparameters (e.g. mini-batch size, learning rate, data augmentation, network structure, etc.) and training process (e.g., masking, training length, optimizer, and selection of input images $I_A$ and $I_B$ in Fig. 2, etc.) as in the original code.

We determine weighting values $\lambda_w$ for $L_{wass}$ as 7.0, 2.0, 3.0, and 0.5 in [3], [6], [9], and [10], respectively. $\lambda_w$ is a weighting factor that balances $L_{origin}$; therefore, there is no significance of the relative sizes of the values. In addition, the vertical and horizontal intervals of the grid points for the sparse sampling shown in Fig. 5 are designed as $n_c = 16$ and $n_r = 4$ to reduce the total number of point clouds for the calculation of the WCL. The effect of $\lambda_w$, $n_c$, and $n_r$ are evaluated in the ablation study later. $\varepsilon$ is set as 0.001 to suppress the approximation in (4) and to stably calculate the WCL.

4.4 Quantitative Analysis

Table 1 displays the widely used seven metrics for the evaluation of depth estimation from monocular camera images. For the leftmost four metrics, smaller is better; for the rightmost three metrics, higher is better. The best value is highlighted in bold.

As summarized in Table 1, the performance of all the baselines can be successfully improved by adding our WCL. The improvement for [3] and [10] is large, whereas it is not quite large for [6] and [9], because they also penalize a geometric inconsistency by their own approach. However, our method has still explicit margins against these baselines. Furthermore, monodepth2[10] + WCL can successfully obtain the best results in most of the metrics.
the ICP loss and the depth consistency loss from the original cost function, respectively. The method of‡ is the result obtained with our proposed method in addition to the baseline one above. For the leftmost four metrics, smaller is better; for the rightmost three metrics, higher is better. † and ‡ indicate removing the ICP loss and the depth consistency loss from the original cost function, respectively. The method of ‡ is evaluated by the author.

Table 2: Ablation study on weighting value $\lambda_c$ (top) and sparse sampling parameters $n_c$ and $n_r$ (bottom) for monodepth2 [10] + WCL. All metrics for evaluation are same as Table 1. Input and output resolutions are 416×128.

| Method      | Image Size | Abs Rel | Sq Rel | RMSE | RMSE log | $\delta < 1.25$ | $\delta < 1.25^2$ | $\delta < 1.25^3$ |
|-------------|------------|---------|--------|------|----------|-----------------|-------------------|-----------------|
| monodepth2  | 640x192    | 0.128   | 1.087  | 5.171| 0.204    | 0.855           | 0.953            | 0.978           |
| + WCL       | 640x192    | 0.127   | 0.995  | 5.039| 0.204    | 0.853           | 0.954            | 0.979           |
| 0.1 16 4    | 640x192    | 0.125   | 0.965  | 5.019| 0.202    | 0.856           | 0.953            | 0.979           |
| 0.5 16 4    | 640x192    | 0.123   | 0.920  | 4.990| 0.201    | 0.858           | 0.953            | 0.980           |
| 1.0 16 4    | 640x192    | 0.128   | 0.993  | 5.158| 0.205    | 0.847           | 0.950            | 0.979           |
| 2.0 16 4    | 640x192    | 0.132   | 1.045  | 5.201| 0.207    | 0.845           | 0.949            | 0.978           |
| 0.5 64 16   | 640x192    | 0.127   | 0.976  | 5.052| 0.203    | 0.852           | 0.953            | 0.979           |
| 0.5 64 8    | 640x192    | 0.126   | 0.957  | 5.081| 0.203    | 0.853           | 0.953            | 0.979           |
| 0.5 32 16   | 640x192    | 0.126   | 0.933  | 5.039| 0.203    | 0.853           | 0.953            | 0.979           |
| 0.5 64 8    | 640x192    | 0.125   | 0.933  | 5.039| 0.203    | 0.853           | 0.953            | 0.979           |
| 0.5 32 4    | 640x192    | 0.125   | 0.938  | 5.006| 0.202    | 0.854           | 0.953            | 0.980           |
| 0.5 16 4    | 640x192    | 0.123   | 0.920  | 4.990| 0.201    | 0.858           | 0.953            | 0.980           |

4.5 Ablation Study

Table 2 shows the results of the ablation study in monodepth2 [10] with our WCL, the upper part is for weighting factor $\lambda_w$ and the lower part is for $n_c$ and $n_r$ of sparse sampling. At the top, $\lambda_w$ is changed under fixed $n_c$ and $n_r$, and at the bottom, $n_c$ and $n_r$ are changed under fixed $\lambda_w$. From the top, it can be seen that increasing the weighting factor can improve the performance of depth estimation, but making it too large worsens the performance, in this case the best result is obtained at $\lambda_w = 0.5$.

In the bottom part, on the other hand, the better performance can be obtained at smaller $n_c$ and $n_r$, because a geometric inconsistency can be accurately penalized. However, since it is difficult to make $n_c$ and $n_r$ even smaller due to the memory limitations of the GPU, we decide $n_c = 16$ and $n_r = 4$ for the evaluation in Table 1.
Figure 6: Qualitative results of our proposed method. The top row displays the input images for the trained neural network to estimate depth images, the second and third rows display the estimated depth images without and with the WCL. The white rectangle box in the depth image highlights the advantages of the WCL.

4.6 Qualitative Analysis

We select two baselines [3, 10] for qualitative analysis and generate the depth images with and without our WCL. The first row in Fig. 6 shows the RGB image as the input for the neural network, second and fourth rows are the estimated depth images by [3] and [10], and third and fifth rows are the estimated depth images by [3] + WCL and [10] + WCL, respectively. In the estimated depth images, we display a white line rectangle to highlight the advantage of our method. Our method can reduce the artifacts in an area far from the camera and/or monochromatic objects.

5 Conclusion

In this paper, we proposed a novel WCL to penalize a geometric inconsistency, for monocular depth estimation. Our proposed approach employed the Wasserstein distance for measuring the consistency between two point clouds from different frames. Our WCL is a smooth and symmetric objective, which can suitably measure the geometric consistency without using any other external and/or non-differentiable libraries. Therefore, the neural network can be trained effectively and efficiently, to obtain highly accurate depth estimation. In the experiment, we employed our proposed WCL in four state-of-the-art baselines and confirmed the benefits of our method with healthy margins.

There are still remaining studies. We need to conduct more evaluations, such as different datasets, larger image sizes [12], and ego-motion estimation.

References

[1] N. Hirose, F. Xia, R. Martín-Martín, A. Sadeghian, and S. Savarese. Deep visual mpc-policy learning for navigation. IEEE Robotics and Automation Letters, 4(4):3184–3191, 2019.

[2] 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), pages 3044–3051. IEEE, 2018.

[3] 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, pages 1851–1858, 2017.

[4] 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), pages 0–0, 2018.
[5] 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, pages 225–234, 2018.

[6] 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, pages 5667–5675, 2018.

[7] 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, pages 2022–2030, 2018.

[8] V. Caser, 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, volume 33, pages 8001–8008, 2019.

[9] 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, pages 8977–8986, 2019.

[10] 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, pages 3828–3838, 2019.

[11] X. Fei, A. Wong, and S. Soatto. Geo-supervised visual depth prediction. IEEE Robotics and Automation Letters, 4(2):1661–1668, 2019.

[12] S. Pillai, R. Ambruș, and A. Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. In 2019 International Conference on Robotics and Automation (ICRA), pages 9250–9256. IEEE, 2019.

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

[14] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In Advances in neural information processing systems, pages 2017–2025, 2015.

[15] G. Peyré, M. Cuturi, et al. Computational optimal transport. Foundations and Trends® in Machine Learning, 11(5-6):355–607, 2019.

[16] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In Advances in neural information processing systems, pages 2292–2300, 2013.

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

[18] 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, pages 2366–2374, 2014.

[19] 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.

[20] 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, pages 1983–1992, 2018.