Depth Completion with Twin Surface Extrapolation at Occlusion Boundaries: Supplementary Material

Saif Imran Xiaoming Liu Daniel Morris
Michigan State University
{imransai, liuxm, dmorris}@msu.edu

Abstract

In the supplementary section, we provide additional insights of our results with SoTA methods, show evidences of boundary outliers on KITTI semi-dense ground-truth and its effect on depth completion performance, and discuss our data generation process in KITTI and Virtual KITTI used for our ablation study in the main paper.

1. Relative Error Maps

It is worthwhile to examine where our method has lower errors in comparison with majority of the SoTA methods which use MSE. For this purpose, we choose the MultiStack method [1] for comparison. we calculate the difference of error maps of Absolute Error, $A(i)$, and Squared Error, $S(i)$, of two methods respectively to show the gains of our method over MultiStack [1]. The error differences are calculated by the following equation:

$$A(i) = |\hat{d}_M(i) - d_t(i)| - |\hat{d}_T(i) - d_t(i)|, \quad (1)$$

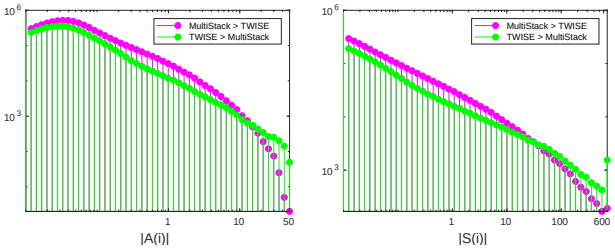
$$S(i) = |\hat{d}_M(i) - d_t(i)|^2 - |\hat{d}_T(i) - d_t(i)|^2, \quad (2)$$

where $\hat{d}_M$ and $\hat{d}_T$ are depth estimates of MultiStack [1] and TWISE respectively. $A(i)$ and $S(i)$ are Absolute Error Difference and Squared Error Difference of pixel i on two competing methods respectively. For a particular pixel, when $A(i)$ and $S(i)$ is $(+)$ve, TWISE is performing better then MultiStack and vice-versa for $(-)$ve values. We note that the errors are evaluated only where there are valid ground-truth pixels.

As shown in Fig. 1, our method wins in substantially more pixels than losing. Errors in our method often comes from few pixels at boundary regions, when a FG depth is erroneously chosen over a BG depth/vice versa; we term them as outliers *e.g.*, see depth error at the traffic sign pixels, edge of tree-trunk etc close to/at the boundary. These outliers with large depth errors are strongly weighted by the RMSE metric, leading to our worse performance on that metric.

Figure 1: Difference of TWISE vs MultiStack [1] in (a) Absolute Error (AE) and (b) Squared Error (SE) respectively. The red indicates the most gain of ours over [1], marked by 'o'; while the blue is vice-versa, marked by 'x'. Zoom in for details.

To further our analysis, we do a statistical evaluation on 200 samples of the validation set (chosen every 5 samples from KITTI's $1,000$ validation set) to confirm that TWISE has better depth estimate on most pixels compared to MultiStack [1] except for few erroneous pixels (outliers) at boundaries (see Fig. 1).



(a) (b)

Figure 2: (a) Magenta is a histogram of absolute error differences $A(i)$ for $A(i) > 0$ (where MultiStack errors > TWISE errors) and green is a histogram of $|A(i)|$ for $A(i) < 0$ (where TWISE errors > MultiStack errors). (b) Corresponding histograms for squared pixel error differences $S(i)$.

We do a histogram binning of $A(i)$ for pixels where $A(i) > 0$ (Multistack > TWISE is equivalent to perfor-

Area	MAE	RMSE	TMAE	TRMSE
Inside Object	196.1	752.3	138.6	327.3
Edge Pixels	731.6	2396.9	304.4	454.6
Whole Image	215.1	880.9	144.6	254.3

Table 1: Error metrics for different image regions on TWISE.

mance gain of TWISE over MultiStack) and of $|A(i)|$ for pixels where $A(i) < 0$ (TWISE > MultiStack is equivalent to performance gain of MultiStack over TWISE). There histograms are plotted together in Fig. 2(a). Analogous histograms are plotted for the squared error difference, $S(i)$, in Fig. 2(b). These histograms show that TWISE has less error than Multi-Stack [1] for most pixels ($\sim 2.70 * 10^6$) compared to just ($\sim 6,100$) pixels where Multi-stack bests TWISE. The average image in this set has $13,500$ pixels where TWISE is better versus 31 pixels where MultiStack is better.

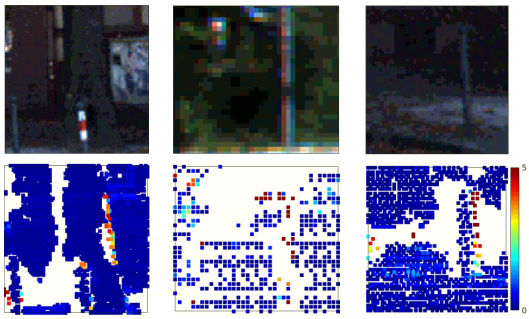


Figure 3: Color images (top) and depth error maps in $0-5$m (bottom).

The reason for large RMSE errors in TWISE is believed to be caused by the outliers (erroneous FG/BG depth selection by TWISE) closer to object boundaries. The outliers are penalized heavily by RMSE metric as opposed to floating depth pixels estimated by MultiStack; as a result, our depth estimate suffers in that metric. As representative examples in Fig. 3, the error maps show depth errors around the boundary, and missing thin objects like poles. The reasoning can be further enhanced by the Tab. 1. In this analysis, we leverage GT semantics provided by KITTI semantic segmentation dataset. In 140 images, FG objects are poles, boundaries, traffic signs, vehicle, person and the rest as background. For each image, we label all pixels where whose distances to object boundareis are less than 3 pixels as edge pixels and the remaining as inside object pixels. Tab. 1 validates substantial larger errors are around boundary.

While outliers can be caused by wrong estimation of foreground/background depth, another important source of outliers is incorrect labelling of ground-truth depths in KITTI. As a result, loss functions that are more sensitive to outliers (i.e. MSE loss) can be negatively influenced by the presence of noise. We highlight the noisy ground-truth labels in KITTI in the next section.

2. Outlier Errors and Analysis on KITTI Semi-Dense GT

In this section we show some evidence of outliers (noisy ground-truth depth) on boundaries of objects in KITTI's semi-dense GT.

Uhrig [2] proposed an approach [2] to generate large-scale semi-dense GT data ($85k$ training images) on realistic outdoor scenes suitable for neural network training. Although the approach is scalable on any dataset, it creates noisy ground-truth depth. Uhrig's [2] analysis shows that the semi-dense GT has larger errors on dynamic objects and large-range pixels. Additionally, we show that it also contains incorrect depth labels on some boundaries of objects. In both (a) and (b) of Fig. 4, we show zoomed in views of how foreground and background depths that are incorrectly spread across the boundaries of the poles, traffic signs, trees etc. of color images.

Our analysis shows that the outliers in the semi-dense GT are caused by a variety of reasons;

- Noisy rotation R, and translation t obtained from the IMU sensor
- Timing synchronization between camera trigger and time taken to spin one lidar revolution
- Consistency Check on Stereo-Global Matching algorithm which introduce boundary artifacts
- Accumulation of lidar points from dynamic objects.

In order to evaluate the depth quality of semi-dense GT, Uhrig [2] used the manually cleaned training set of 2015 KITTI stereo benchmark as reference data. The depth evaluation is done in pixel units. We realize that it is equally important to evaluate the semi-dense ground-truth depths in metric units to notice the *effect of boundary outliers* on semi-dense ground-truth depth metric performance. We translate the error in pixel units to error in metric units in Tab. 2, by converting the ground-truth disparity to depth using KITTI's provided intrinsics. It shows the noisy semi-dense ground-truth depths suffering from boundary noise and dynamic objects can also have significant errors in metric units. It is also a possible indication that lowering the RMSE error in semi-dense GT might result in learning the noise inherent in semi-dense ground-truth.

3. Sparse Patterns in KITTI

In the main paper, we show the improved generalizability of TWISE over other SoTA methods in terms of sparsity. In

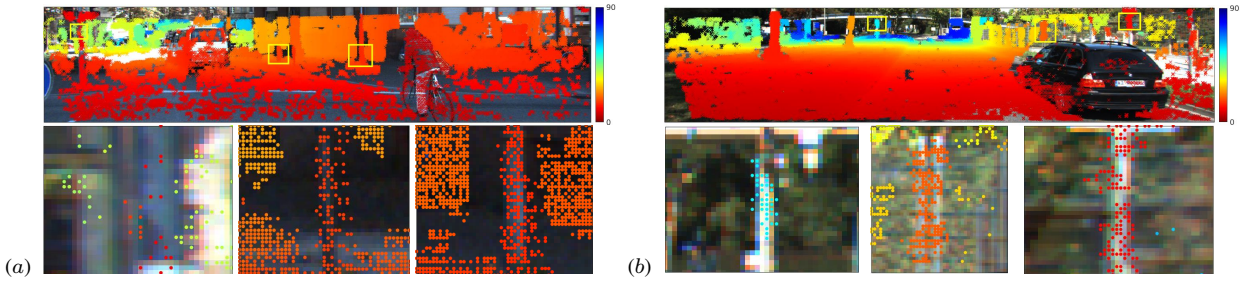


Figure 4: Semi-dense GT depths overlaid on color images. Zoom-in views show foreground/background depths are incorrectly spread (dilated/constricted) across boundaries of poles, traffic signs etc. visible in color images.

MAE (in pixel)	RMSE (in pixel)	KITTI Outliers*	MAE (in cm)	RMSE (in cm)
0.35	0.84	0.31	38.6	94.1

Table 2: Relation between Disparity Error and Depth Error in metric units (cm). Note that KITTI Outliers are defined by: > 3 pix disparity error and 5% error.

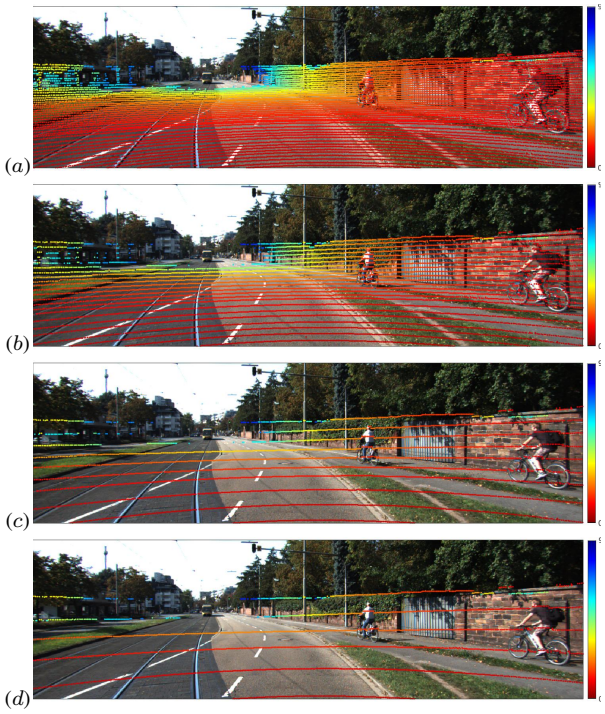


Figure 5: KITTI sparse patterns of (a) $64R$, (b) $32R$, (c) $16R$, and (d) $8R$ subsampled lidar respectively overlaid on a color image.

this section, we explain how sparsity is created from $64R$ lidar in KITTI. Ma *et al.* [3] reported improved performance with uniform subsampling from KITTI's ground-truth data. But in real scenarios, sparse sensors such as lidar often generate non-uniform, structured patterns. We simulate lower resolution lidars by subsampling 32R, 16R, 8R rows from 64R lidar (depth acquisition sensor used by KITTI). The different sparse patterns can be seen in Fig. 5. We subsample the points based on selecting a subset of evenly spaced rows of 64R raw data provided by KITTI (split based on the azimuth angle in the lidar space) and then projecting the points into the image.

4. Network Architecture

In the main paper, we mentioned that we used the network of Li *et al.* [1] as a backbone network for TWISE. The only modification we made are at the last layer of the network, where we used three channels representing d_1 (foreground estimate), d_2 (background estimate), and σ (see Fig. 7). We repeat this strategy in the hourglass networks in all the three multi-resolution levels. Please see [1] for more details of the network.

5. Additional VKITTI Resutls

5.1. VKITTI Results on Different Weathers

The high-resolution color features is an important cue for FG/BG selection in TWISE. We also analyze the effect of different weather conditions that can deteriorate high-resolution boundary cues from color in Tab. 3. In this study, we found that model trained on 'clone' set is evaluated on different weather conditions in VKITTI.T The performance is largely maintained, with minor degradations in fog and rain. It shows although the low-quality RGB (low contrast, shadows, fog, rain etc) might create ambiguity and the blending coefficient fail to correctly select FG/BG, it is possible to detect boundary information using sufficient training examples.

5.2. Creating Semi-Dense and Sparse Depth from Dense VKITTI GT

In the main paper, we performed an ablation study on Virtual KITTI [4] (VKITTI) using semi-dense and sparse samples created from dense VKITTI depth maps. We created

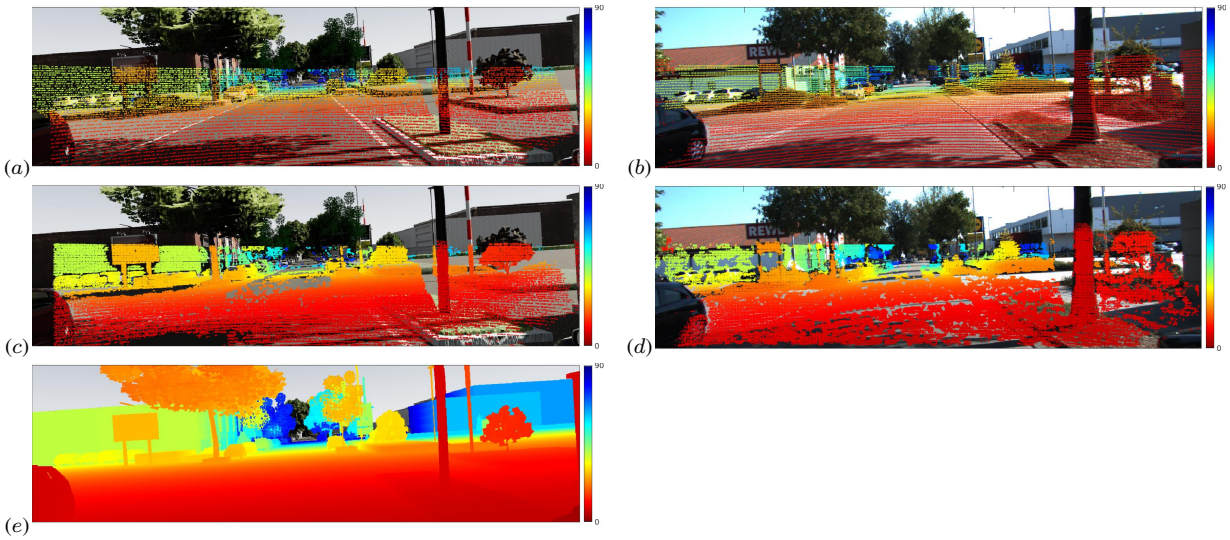


Figure 6: Visual examples of (a) sparse depth, (c) semi-dense depth and (e) dense depth of virtual KITTI. (b) and (d) shows sparse depth and semi-dense GT of KITTI respectively (shown for comparison with VKITTI data).

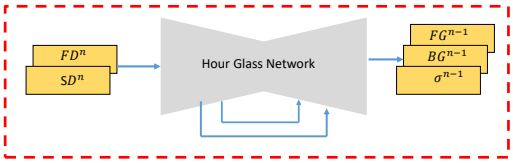


Figure 7: Incorporating 3-channel at the output of the Hourglass network used in [1]. SD^n and FD^n are the sparse inputs and fused depth obtained from FG^n, BG^n, and σ^n at multi resolution scale n respectively.

RGB Mode	MAE	RMSE	TMAE	TRMSE
Clone	12.71	126.40	5.22	16.67
Morning	12.99	130.90	5.17	16.60
Fog	13.19	131.97	5.15	16.74
Sunset	12.77	129.50	5.10	16.50
Rainy	13.08	132.09	5.17	16.67
OverCast	12.48	126.82	5.08	16.47

Table 3: VKITTI Results on different weather conditions

semi-dense VKITTI to simulate outlier noise similar to that existing in real KITTI dataset. In this section, we discuss the data generation process in detail and show some visual examples of how the sparse depth/semi-dense compares with sparse/semi-dense gt of KITTI dataset in Fig. 6.

The dense ground-truth depth maps from VKITTI contains accurate depth on object discontinuities. Using this as a reference, we subsampled the ground-truth depth maps. Instead of uniformly subsampling the GT depth, we subsample

Dataset	R, t	Outliers%	Pix. Coverage%	MAE (cm)	RMSE (cm)
KITTI	IMU	4.4	16	38.6	94.1
VKITTI	Clean R, t	3.0	20	19.3	128.96
	Noisy R, t	4.1	18	29.3	145.18

Table 4: Comparison of VKITTI semi-dense errors with KITTI semi-dense GT errors. Higher errors in RMSE in the VKITTI dataset is due to dense depth pixels at far-away points, contrary to KITTI's stereo benchmark data which is sparse.

the lidar in the azimuth-elevation coordinates to make the input sparse depth resemble structured patterns found in original lidar (see (a) and (b) of Fig. 6). The subsampled depth from the left camera is then projected to the right camera, and vice versa to simulate lidar points projected onto images in real-world scenes. For supervision, GT depth beyond 90m are suppressed to simulate lidar points with no returns (see (e) of Fig. 6). In addition to supervision using clean ground-truth present, we also perform supervision on Semi-Dense GT of VKITTI (Fig. 10 of the main paper) created by simulating outliers existing in original KITTI dataset [2]. In the KITTI dataset, semi-dense GT is created by accumulating lidar points from $+/-5$ frames from the reference frame. We follow the similar procedure as followed by [2] when creating semi-dense GT. Additionally, we add Gaussian noise to model noisy R, t from the IMU sensor to simulate noisy semi-dense GT. Refer to Fig. 6 for a comparison between semi-dense VKITTI and semi-dense KITTI (see (c) and (d) of Fig. 6).

5.3. Relation to KITTI GT by Outliers

We define outliers as pixels having depth errors greater than 1m, contrary to KITTI outliers in Tab. 2 which define errors in pixel units. Evaluated on KITTI's 2015 stereo benchmark depth data, we found outliers of KITTI's semi-dense ground-truth at 4.4% of the inlier depths. We created outliers in semi-dense VKITTI by introducing Gaussian noise in VKITTI's extrinsics. See Tab. 4 for a metric comparison with outliers. Tab. 4 shows that, as we add noisy in R, t, the semi-dense GT of VKITTI is more comparable to KITTI semi-dense GT.

6. Video

We provide a video in the supplementary material. The video shows point-cloud rendered from estimated depth maps of TWISE and MultiStack [1]. It shows point-cloud generated from MultiStack contains significantly more mixed depth pixels (compare the floating depth pixels in the pointclouds) compared to TWISE.

References

[1] A. Li, Z. Yuan, Y. Ling, W. Chi, s. zhang, and C. Zhang, "A multi-scale guided cascade hourglass network for depth completion," in *IEEE Workshop Application Computer Vision (WACV)*, 2020.

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

[3] F. Ma, G. V. Cavalheiro, and S. Karaman, "Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, IEEE, 2019, pp. 3288–3295.

[4] Y. Cabon, N. Murray, and M. Humenberger, "Virtual kitti 2," 2020.