工作总结与计划

一、 工作总结（按时间划分）

日期	工作内容			完成情况及主要问题
	上午	下午	晚上	
星期一	学习算法	读论文	总结算法	阅读上周论文使用的网络，学习网络使用的 plane-sweep 算法原理
星期二	读论文	读论文	读论文	完成论文的阅读
星期三	阅读有关书籍	学习 pytorch	开会	对研究方向的研究思路和关键问题有了初步的了解
星期四	读论文	读论文	读论文	选择单目深度估计的论文进行阅读
星期五	读参考文献	休息	复读论文	对论文中描述简述的内容通过阅读参考文献进行理解
星期六	复读论文	写周报	写周报	复读论文和完成周报

二、下一步计划（按任务划分）

编号	工作内容	目标	相关配合
1	阅读单目深度估计的论文	12 月 11 日前完成	无
2	准备毕设开题报告	12 月 25 日前完成	无
3			
4			
5			
主要风险	无		

三、个人分析与总结

内容提要	
1	进度方面：本周能有效按照预期计划完成目标
2	课题方面：通过交流对课题的关键问题有了一定认识
3	其他思考：在阅读论文外要学深度学习知识，提高英语水平
4	

四、论文总结

论文标题	DPSNet: End -to -end Deep Plane Sweep Stereo
作者及单位	Sunghoon Im[*1], Hae-Gon Jeon[2], Stephen Lin[3], In So Kweon[1] [1] KAIST, [2] Carnegie Mellon University, [3] Microsoft Research Asia
论文出处	2019-ICLR
创新点提炼	（1） Propose a new End-to-end Multiview Matching method （2） Design a Deep Cost Volume Aggregation method （3） All networks are inspired by traditionalmulti-view stereo methods Plane Sweeping , Cost Volume filtering
个人想法	

论文方法及结论：

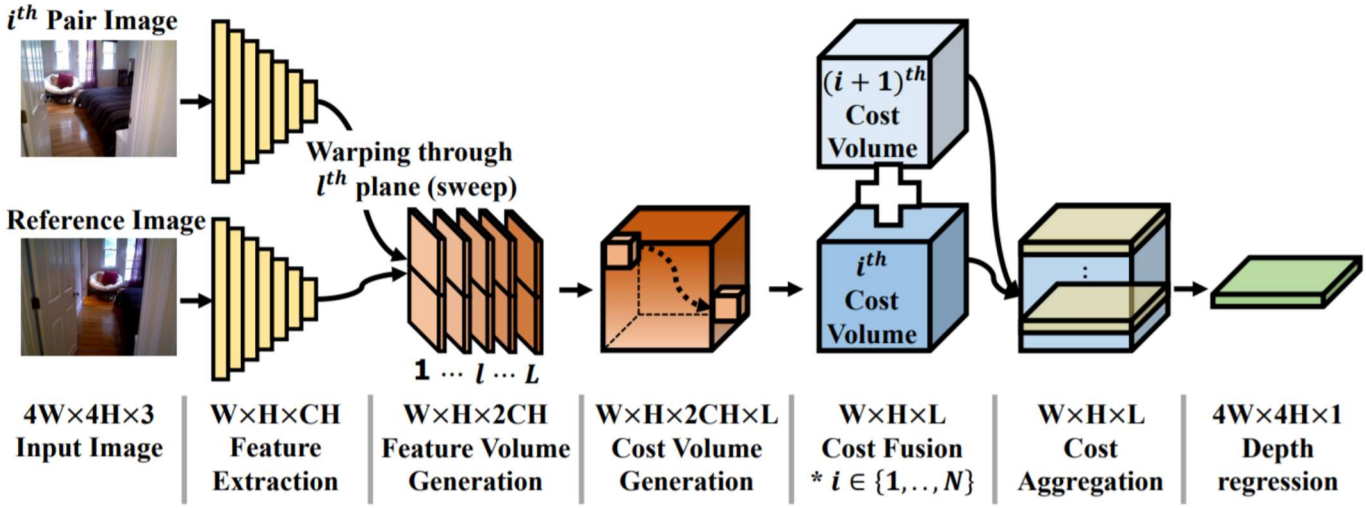


Figure 1. Overview of the DPSNet pipeline.

Deep Plane Sweep Network（DPSNet）受传统的多视图立体实践的启发，用于密集深度估计，它包括四个部分：特征提取，成本量（cost volume）生成，代价聚合（cost aggregation）和深度图回归。

1. Multi-scale Feature Extraction

将参考图像和目标图像通过七个卷积层（除第一层除为7 × 7的卷积核，其余层具有3 × 3的卷积核）进行编码，并使用包含四个固定大小的平均池化（16 × 16; 8 × 8; 4 × 4; 2 × 2）的空间金字塔池化（spatial pyramid pooling, SPP）从这些图像中提取层级上下文信息。

在将层级的上下文信息上采样到与原始特征图相同的大小后，将所有特征图连接起来，然后将它们传递到 2D 卷积层中。此过程为所有输入图像生成32通道特征表示，接下来将其用于构建 cost volume。

2. Cost Volume Generation

1) 设置垂直于z轴的虚拟平面的数量
2) 在反深度（inverse-depth）空间均匀采样 $d_l = \frac{(L \times d_{min})}{l}, (l = 1, \ldots, L)$
3) 使用相机的内参 $\mathbf{K}$ 和外参 $[\mathbf{R}|\mathbf{t}]$ 将所有成对的特征 $\mathcal{F}_i$ 变形为参考特征的坐标 $\tilde{\mathcal{F}}_{il}(u) = \mathcal{F}_i(\tilde{u}_l), \quad \tilde{u}_l \sim \mathbf{K}[\mathbf{R}_i \mid \mathbf{t}_i] \begin{bmatrix} (\mathbf{K}^{-1}u)d_l \\ 1 \end{bmatrix}$
4) 给定4D volume，在串联特征上使用一系列 3D 卷积学习 cost volume 的生成

3. Cost Aggregation

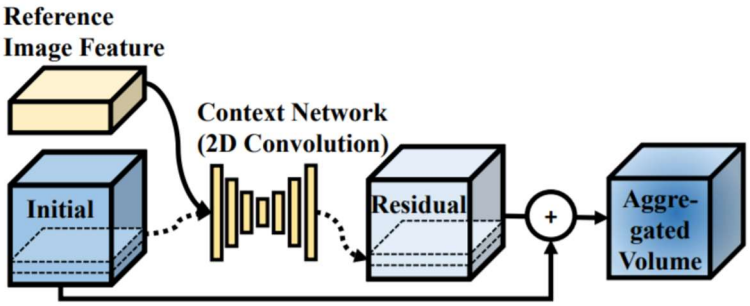


Figure 2. Illustration of context-aware cost.

受传统成 cost volume 过滤的启发，我们在端到端学习过程中引入了一种上下文感知的代价聚合方法（context-aware cost aggregation）。上下文网络获取 cost volume 的每个切片以及从上一步中提取的参考图像特征，然后输出优化后的切片。对所有切片进行相同的操作。然后，通过将初始量和剩余量相加来获得最终 cost volume。

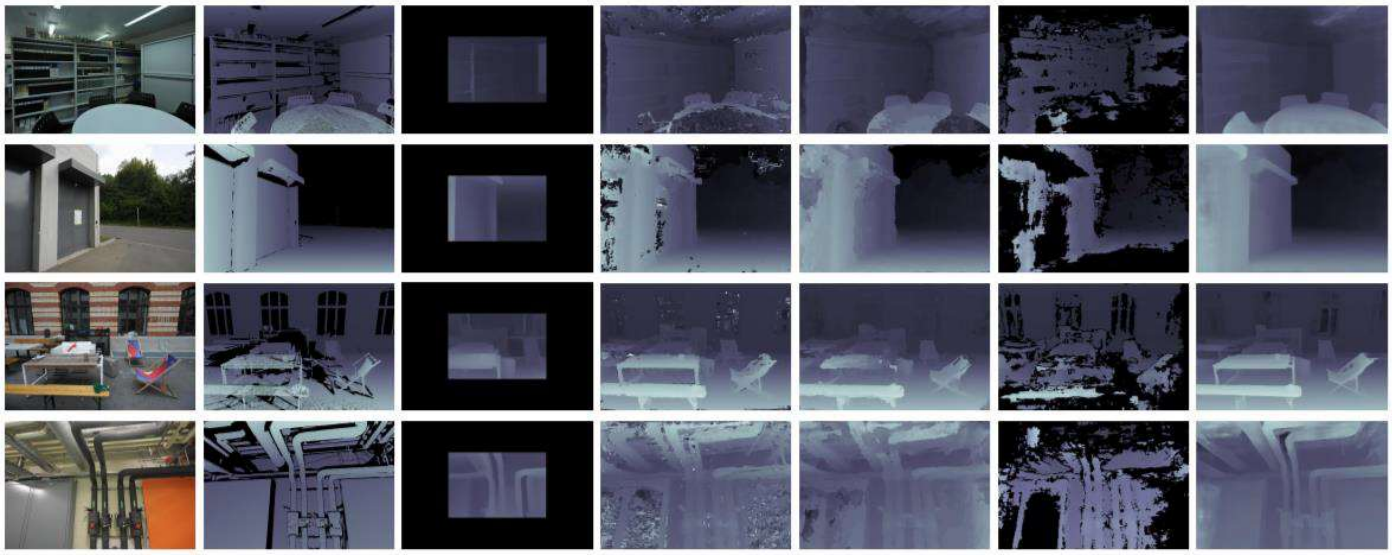
4. Depth Regression

1) 回归连续深度值

2) 每个标签l的概率通过 softmax 运算 $\sigma(\cdot)$ 从预测成本 c_l 计算

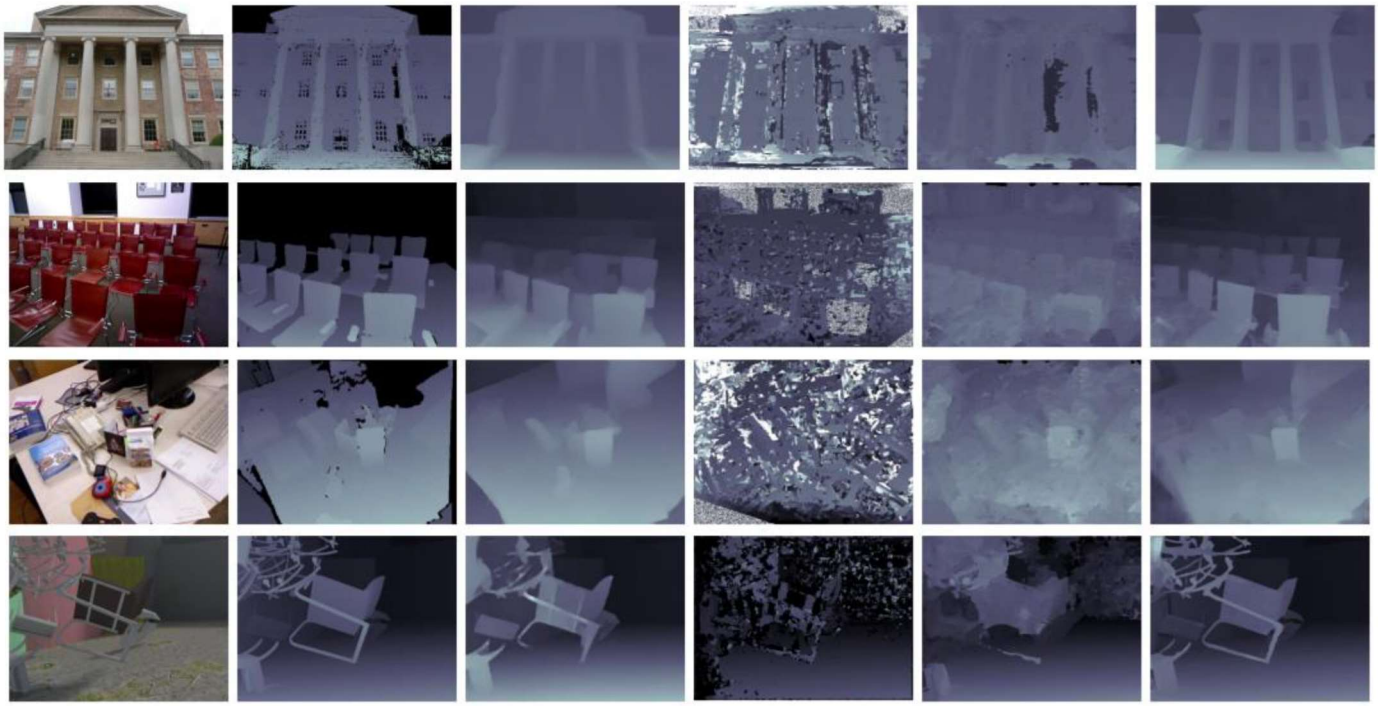
3) 计算为每个标签 l 的加权总和 $\tilde{d} = \frac{L \times d_{min}}{\tilde{l}}$, $\tilde{l} = \sum_{l=1}^{L} l \times \sigma(c_l)$

Results from ETH3D datasets



Method	Compl -eteness	Error metric							Accuracy metric ($\delta < \alpha^t$)		
		Geo.	Photo.	A. Rel	A. diff	Sq Rel	RMSE	Rlog	α	α^2	α^3
COLMAP filter	71 %	0.007	0.178	0.045	0.033	**0.293**	0.619	0.123	0.965	0.978	0.986
COLMAP	100 %	0.046	0.218	0.324	0.615	36.71	2.370	0.349	**0.865**	0.903	0.927
DeMoN	100 %	0.045	0.288	0.191	0.726	0.365	1.059	0.240	0.733	0.898	0.951
DeepMVS	100 %	0.036	0.224	0.178	0.432	0.973	1.021	0.245	0.858	0.911	0.942
MVSNET filter	77 %	0.067	**0.179**	0.357	0.766	1.969	1.325	0.423	0.706	0.779	0.829
MVSNET	100 %	0.077	0.218	1.666	2.165	13.93	3.255	0.824	0.555	0.628	0.686
Ours	100 %	**0.034**	0.202	**0.099**	**0.365**	0.204	**0.703**	**0.184**	0.863	**0.938**	**0.963**

Results from MVS, SUN3D, RGBD, Scenes11 datasets



Data -sets	Method	Error metric					Accuracy metric ($\delta < \alpha^t$)		
		Abs Rel	Abs diff	Sq Rel	RMSE	RMSE log	α	α^2	α^3
MVS	COLMAP	0.3841	0.8430	1.257	1.4795	0.5001	0.4819	0.6633	0.8401
	DeMoN	0.3105	1.3291	19.970	2.6065	0.2469	0.6411	0.9017	0.9667
	DeepMVS	0.2305	0.6628	0.6151	1.1488	0.3019	0.6737	0.8867	0.9414
	Ours	**0.0689**	**0.2290**	**0.0930**	**0.6725**	**0.1542**	**0.8868**	**0.9441**	**0.9682**
SUN3D	COLMAP	0.6232	1.3267	3.2359	2.3162	0.6612	0.3266	0.5541	0.7180
	DeMoN	0.2137	2.1477	1.1202	2.4212	0.2060	0.7332	0.9219	0.9626
	DeepMVS	0.2816	0.6040	0.4350	0.9436	0.3633	0.5622	0.7388	0.8951
	Ours	**0.1470**	**0.3234**	**0.1071**	**0.4269**	**0.1906**	**0.7892**	**0.9317**	**0.9672**
RGBD	COLMAP	0.5389	0.9398	1.7608	1.5051	0.7151	0.2749	0.5001	0.7241
	DeMoN	0.1569	1.3525	0.5238	1.7798	**0.2018**	0.8011	0.9056	0.9621
	DeepMVS	0.2938	0.6207	0.4297	0.8684	0.3506	0.5493	0.8052	0.9217
	Ours	**0.1538**	**0.5235**	**0.2149**	**0.7226**	0.2263	**0.7842**	**0.8959**	**0.9402**
Scenes11	COLMAP	0.6249	2.2409	3.7148	3.6575	0.8680	0.3897	0.5674	0.6716
	DeMoN	0.5560	1.9877	3.4020	2.6034	0.3909	0.4963	0.7258	0.8263
	DeepMVS	0.2100	0.5967	0.3727	0.8909	0.2699	0.6881	0.8940	0.9687
	Ours	**0.0558**	**0.2430**	**0.1435**	**0.7136**	**0.1396**	**0.9502**	**0.9726**	**0.9804**

论文标题	Self-supervised Monocular Trained Depth Estimation using Self-attention and Discrete Disparity Volume
作者及单位	Adrian Johnston Gustavo Carneiro Australian Institute for Machine Learning School of Computer Science, University of Adelaide
论文出处	2020-CVPR
创新点提炼	（1） 采用 self-attention 和 discrete disparity volume 为 KITTI 2015 和 Make3D 产生了最佳的自监督单目深度估计结果。在实验中表明，该方法能够使用自监督的立体训练来缩小和全监督的深度估计的差距。 （2） 利用语义分割网络体系结构的最新进展，能够在一台 GPU 机器上训练更大的模型。
个人想法	

论文方法及结论：

1. 论文提出的问题

- 对于全监督的方法，由于传感器噪声和操作能力有限，因此收集准确且庞大的真实数据集是一项艰巨的任务；
- 自监督的单目深度估计需要共同估计深度和自我运动以最小化光度重投影损失函数。姿势估计器模型引入的任何噪声都会降低在单目序列上训练的模型的性能，从而导致深度较大的估计误差。此外，自监督的单目模型假设在静态场景中使用移动摄像机，这使得单目模型将与移动物体有关的像素预测为“孔”。

2. 解决的办法

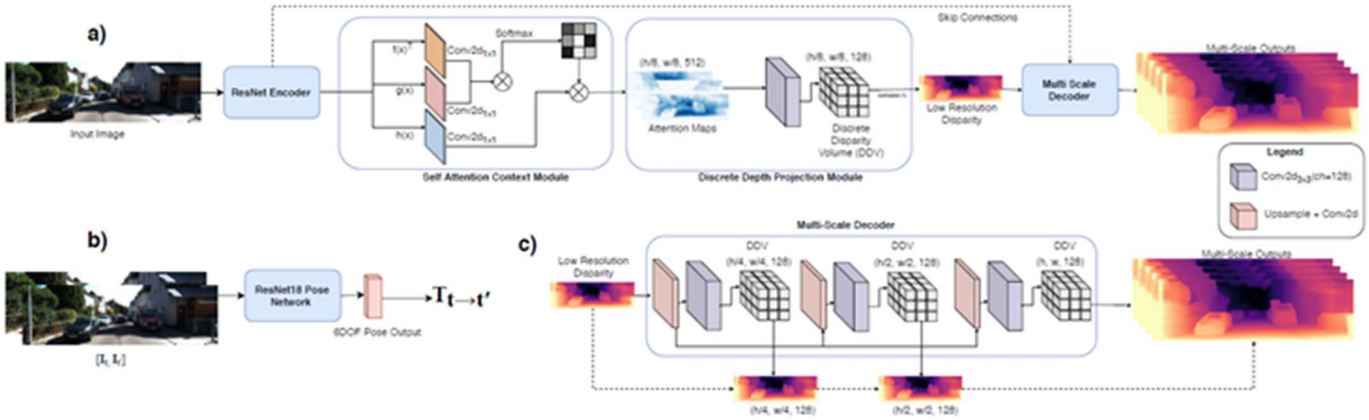


Figure 3. **Overall Architecture** The image encoding processes is highlighted in part a). The input monocular image is encoded using a ResNet encoder and then passed through the Self-Attention Context Module. The computed attention maps are then convolved with a 2D convolution with the number of output channels equal to the number dimensions for the Discrete Disparity Volume (DDV). The DDV is then projected into a 2D depth map by performing a softargmax across the disparity dimension resulting in the lowest resolution disparity estimation (Eq. 4). In part b) the pose estimator is shown, and part c) shows more details of the Multi-Scale decoder. The low resolution disparity map is passed through successive blocks of UpConv (nearest upsample + convolution). The DDV projection is performed at each scale, in the same way as in the initial encoding stage. Finally, each of the outputs are upsampled to input resolution to compute the photometric reprojection loss.

本文中提出了两个新的想法来改进自监督的单目深度估计：self-attention 和 discrete disparity volume。self-attention 模块与当前使用的局部 2D 和 3D 卷积无法探索全局上下文形成对比。如先前由全监督的深度估计方法所证明的，discrete disparity volume 使得能够估计更鲁棒和更锐利的深度估计。此外，该方法可以利用 discrete disparity volume。self-attention 估计像素级深度不确定性。

如图 3 所示，首先使用 ResNet-101 对 RGB 图像（I: Ω → R^3，where Ω denotes the image lattice of height H and width W）进行编码，即 $X = resnet_\theta(I_t)$，with X: $\Omega_{1/8} \rightarrow R^M$，,M denoting the number of channels at the output of the ResNet。Resnet 的输出被用于 self-attention 模块，如下图所示

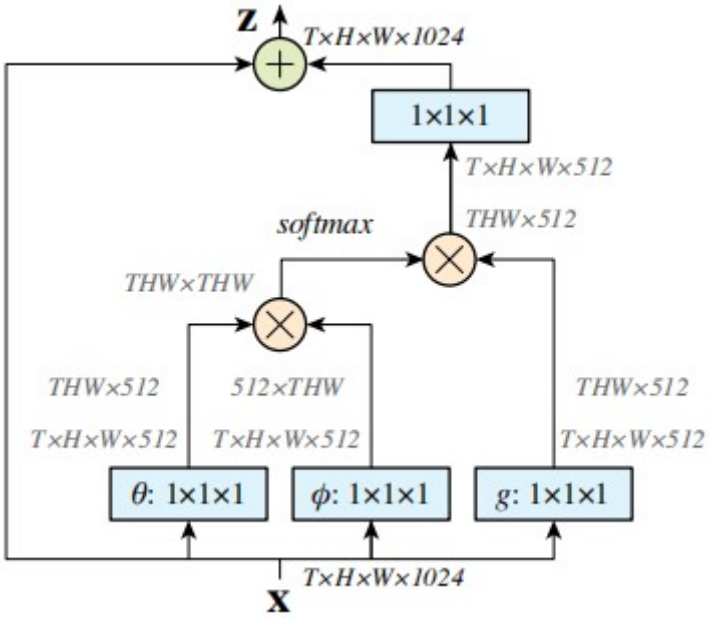


Figure 4. A spacetime **non-local block**. The feature maps are shown as the shape of their tensors, e.g., T×H×W×1024 for 1024 channels (proper reshaping is performed when noted). "⊗" denotes matrix multiplication, and "⊕" denotes element-wise sum. The softmax operation is performed on each row. The blue boxes denote 1×1×1 convolutions. Here we show the embedded Gaussian version, with a bottleneck of 512 channels. The vanilla Gaussian version can be done by removing θ and φ, and the dot-product version can be done by replacing softmax with scaling by 1/N.

首先形成 query, key and value results,

$$f\big(X(\omega)\big) = W_f X(\omega)$$
$$g\big(X(\omega)\big) = W_g X(\omega)$$
$$h\big(X(\omega)\big) = W_h X(\omega)$$

query 和 key 的值一起得到

$$S_\omega = softmax\left(f\big(X(\omega)\big)^T g(X)\right)$$

self-attention map 可被构建得到

$$\mathbf{A}(\omega) = \sum_{\tilde{\omega}\in\Omega_{1/8}} h(\mathbf{X}(\tilde{\omega})) \times \mathbf{S}_\omega(\tilde{\omega})$$

DDV 可描述为 $D_{1/8}(\omega) = conv_{3\times3}\big(A(\omega)\big)$，低分辨率的 disparity map 可通过如下计算获得

$$\sigma\big(\mathbf{D}_{1/8}(\omega)\big) = \sum_{k=1}^{K} \mathrm{softmax}\big(\mathbf{D}_{1/8}(\omega)[k]\big) \times di\,\mathrm{sparity}(k)$$

由于低分辨率视差图产生的结果不明确，本文使用 Godard 等人提出的多尺度策略。低分辨率图是多尺度解码器的第一步，该解码器由 upconv 运算符的三个附加阶段组成（即，nearest upsample + convolution），它们从 ResNet 编码器接收相应分辨率的跳过连接，如图 3 所示。众所周知，编码层和关联的解码层之间的这些跳过连接可在最终深度输出中保留高级信息。在每种分辨率下，形成一个新的 DDV，该 DDV 用于计算该特定分辨率下的视差图。

姿势估计，它会采用两个在两个不同时间步长记录的图像，然后返回相对变换，$T_{t\to t'} = p_\phi\left(I_t, I_{t'}\right)$

3. 实验结果结论

Method	Train	Abs Rel	Sq Rel	RMSE	RMSE log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Eigen [9]	D	0.203	1.548	6.307	0.282	0.702	0.890	0.890
Liu [30]	D	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Klodt [24]	D*M	0.166	1.490	5.998	-	0.778	0.919	0.966
AdaDepth [38]	D*	0.167	1.257	5.578	0.237	0.771	0.922	0.971
Kuznietsov [25]	DS	0.113	0.741	4.621	0.189	0.862	0.960	0.986
DVSO [55]	D*S	0.097	0.734	4.442	0.187	0.888	0.958	0.980
SVSM FT [33]	DS	0.094	0.626	4.252	0.177	0.891	0.965	0.984
Guo [15]	DS	0.096	0.641	4.095	0.168	0.892	0.967	0.986
DORN [10]	D	**0.072**	**0.307**	**2.727**	**0.120**	**0.932**	**0.984**	**0.994**
Zhou [62]†	M	0.183	1.595	6.709	0.270	0.734	0.902	0.959
Yang [57]	M	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Mahjourian [34]	M	0.163	1.240	6.220	0.250	0.762	0.916	0.968
GeoNet [58]†	M	0.149	1.060	5.567	0.226	0.796	0.935	0.975
DDVO [53]	M	0.151	1.257	5.583	0.228	0.810	0.936	0.974
DF-Net [63]	M	0.150	1.124	5.507	0.223	0.806	0.933	0.973
LEGO [56]	M	0.162	1.352	6.276	0.252	-	-	-
Ranjan [43]	M	0.148	1.149	5.464	0.226	0.815	0.935	0.973
EPC++ [32]	M	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Struct2depth '(M)' [3]	M	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Monodepth2 [14]	M	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Monodepth2 (1024 × 320)[14]	M	0.115	0.882	4.701	0.190	0.879	0.961	**0.982**
Ours	M	**0.106**	**0.861**	**4.699**	**0.185**	**0.889**	**0.962**	**0.982**
Garg [11]†	S	0.152	1.226	5.849	0.246	0.784	0.921	0.967
Monodepth R50 [13]†	S	0.133	1.142	5.533	0.230	0.830	0.936	0.970
StrAT [36]	S	0.128	1.019	5.403	0.227	0.827	0.935	0.971
3Net (R50) [42]	S	0.129	0.996	5.281	0.223	0.831	0.939	0.974
3Net (VGG) [42]	S	0.119	1.201	5.888	0.208	0.844	0.941	**0.978**
SuperDepth + pp [41] (1024 × 382)	S	0.112	0.875	4.958	0.207	0.852	0.947	0.977
Monodepth2 [14]	S	0.109	0.873	4.960	0.209	0.864	0.948	0.975
Monodepth2 (1024 × 320)[14]	S	**0.107**	**0.849**	**4.764**	**0.201**	**0.874**	**0.953**	0.977
UnDeepVO [28]	MS	0.183	1.730	6.57	0.268	-	-	-
Zhan FullNYU [60]	D*MS	0.135	1.132	5.585	0.229	0.820	0.933	0.971
EPC++ [32]	MS	0.128	0.935	5.011	0.209	0.831	0.945	0.979
Monodepth2[14]	MS	**0.106**	0.818	4.750	0.196	0.874	0.957	0.979
Monodepth2(1024 × 320)[14]	MS	**0.106**	**0.806**	**4.630**	**0.193**	**0.876**	**0.958**	**0.980**

Table 1. **Quantitative results.** Comparison of existing methods to our own on the KITTI 2015 [12] using the Eigen split [8]. The Best results are presented in **bold** for each category, with second best results underlined. The supervision level for each method is presented in the *Train* column with; D – Depth Supervision, D* – Auxiliary depth supervision, S – Self-supervised stereo supervision, M – Self-supervised mono supervision. Results are presented without any post-processing [13], unless marked with – + pp. If newer results are available on github, these are marked with – †. Non-Standard resolutions are documented along with the method name. Metrics indicated by red: *lower is better*, Metrics indicated by blue: *higher is better*

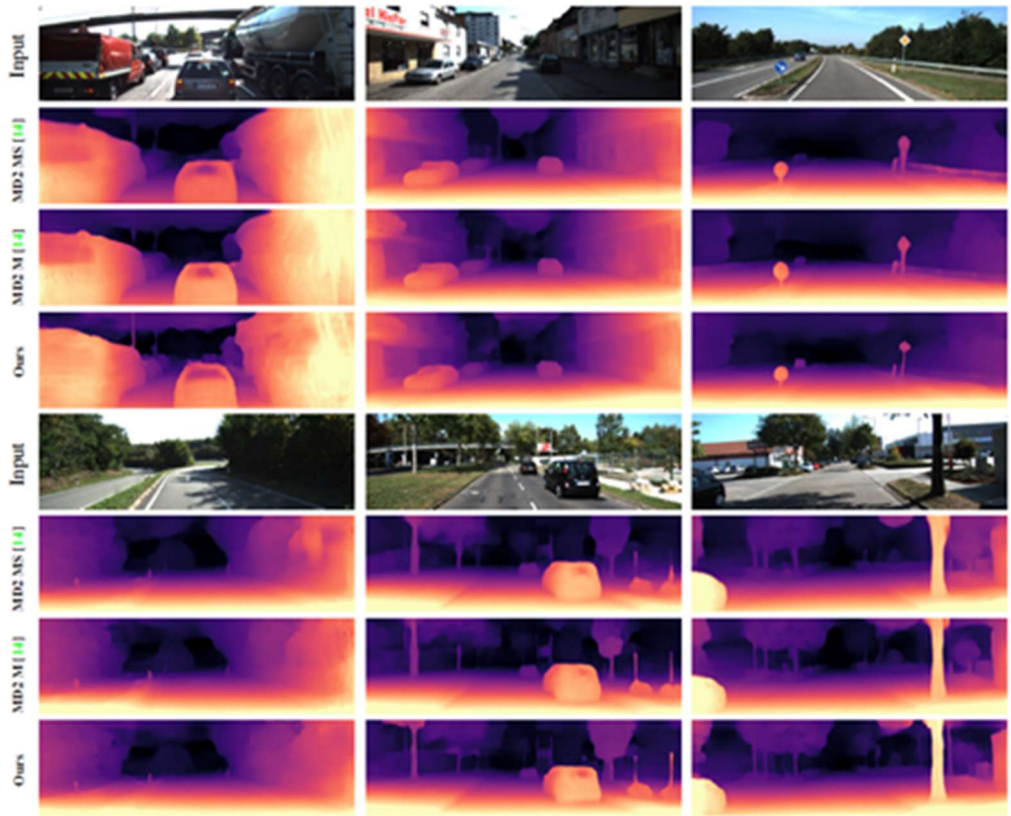


Figure 3. **Qualitative results on the KITTI Eigen split [8] test set.** Our models perform better on thinner objects such as trees, signs and bollards, as well as being better at delineating difficult object boundaries.

4. 存在的问题

暂无