



DIG

P3Depth: Monocular Depth Estimation with a Piecewise Planarity Prior (CVPR 2023)

2023.08.17

① **Learning to Identify Seed Pixels**

② **Mean Plane Loss.**

2

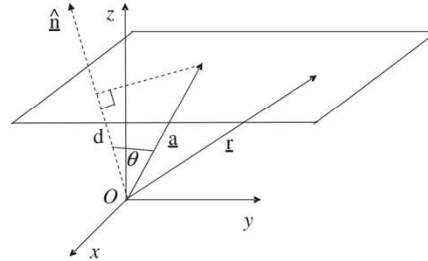
Depth Plane的CoEfficient借鉴于GeoLayOut.

Offset Vector借鉴于Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth.

where $\hat{n} (= \frac{\underline{b} \times \underline{c}}{|\underline{b} \times \underline{c}|})$ is the **unit** vector perpendicular to the plane.

5.1.1 Plane from vector to Cartesian form

- ▶ $(\underline{r} - \underline{a}) \cdot \hat{n} = 0$ gives $\underline{r} \cdot \hat{n} = \underline{a} \cdot \hat{n}$
- ▶ Note that
 $d = a \cos \theta = \underline{a} \cdot \hat{n}$ is the perpendicular distance of the plane to the origin.
- ▶ Also we write
 $\hat{n} = l\hat{i} + m\hat{j} + n\hat{k}$,
 where (l, m, n) are defined as the *direction cosines* of the normal to the plane.
- ▶ Finally we write the general vector \underline{r} as (x, y, z)
- ▶ This gives the plane in Cartesian representation as



$$\underline{r} \cdot \hat{n} = lx + my + nz = d$$

这里的法向量是具有单位长度的法向量，原文中也没有说明（☺），然后这个P不应该是Point的意思，应该是Poositional Vector的意思。

Cartesian representation ()

fxfy是焦距，uovo是光轴和成像平面的交点，d是平面距离原点的距离。

$$Z = D(u, v), \quad X = \frac{Z(u - u_0)}{f_x}, \quad Y = \frac{Z(v - v_0)}{f_y}. \quad (2)$$

$$\mathbf{n} \cdot \mathbf{P} + d = 0, \text{ where } \mathbf{n} = (a, b, c)^T \quad \frac{1}{Z} = \underbrace{\frac{-a}{f_x d}}_{\hat{\alpha}} u + \underbrace{\frac{-b}{f_y d}}_{\hat{\beta}} v + \underbrace{\frac{1}{d} \left(\frac{a}{f_x} u_0 + \frac{b}{f_y} v_0 - c \right)}_{\hat{\gamma}}.$$

$$\rho = \sqrt{\hat{\alpha}^2 + \hat{\beta}^2 + \hat{\gamma}^2} \text{ and normalizing } \alpha = \frac{\hat{\alpha}}{\rho}, \beta = \frac{\hat{\beta}}{\rho} \text{ and } \gamma = \frac{\hat{\gamma}}{\rho} \text{ into}$$

$$Z = [(\alpha u + \beta v + \gamma)\rho]^{-1}.$$

4

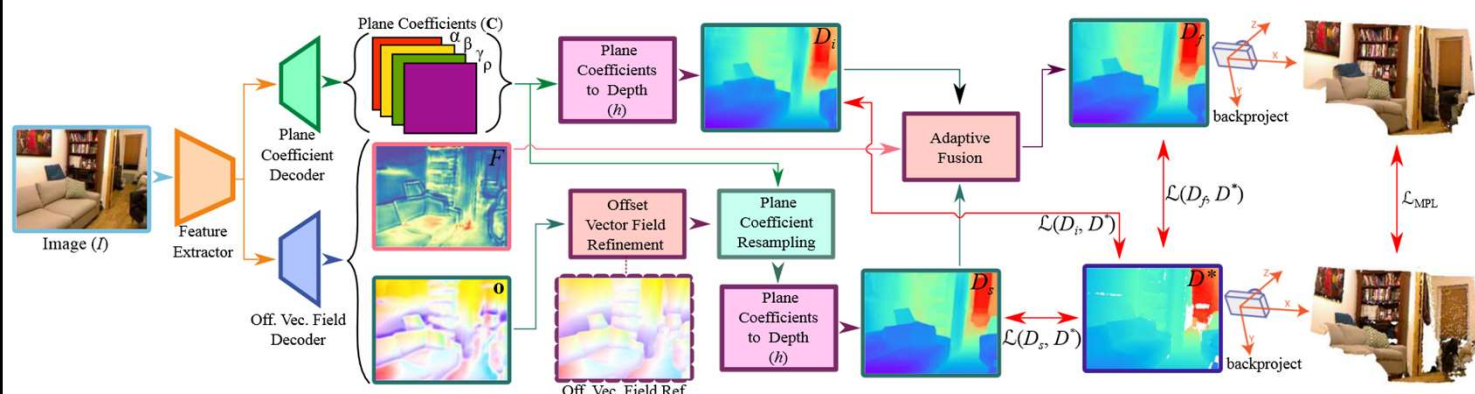
这里的法向量是具有单位长度的法向量，原文中也没有说明（☹️），然后这个P不应该是Point的意思，应该是Positional Vector的意思。

Cartesian representation

fxfy是焦距，uovo是光轴和成像平面的交点，d是平面距离原点的距离。

$$\mathbf{C} = (\alpha, \beta, \gamma, \rho)^T \quad h : (\mathbf{C}(u, v), u, v) \rightarrow D_i(u, v) \quad Z = h(\mathbf{C}, u, v)$$

$$f_\theta = h \circ (\mathbf{g}_\theta, \mathbf{p}), \text{ where } \mathbf{g}_\theta : I(u, v) \rightarrow \mathbf{C}(u, v)$$



与直接预测深度相比，预测平面系数作为中间输出并不具有直接优势。然而，描绘同一三维平面的两个像素具有相同的参数 \mathbf{C} ，但深度通常不同。这一事实是网络下一部分的核心，它允许通过选择性地从种子像素引导平面系数来预测深度。

先验：对于具有关联 3D 平面的每个像素 \mathbf{p} ，在 \mathbf{p} 的邻域中存在一个种子像素 \mathbf{q} ，该种子像素 \mathbf{q} 也与与 \mathbf{p} 相同的平面关联。总的， \mathbf{p} 可能存在多个种子像素或没有种子像素

先验：对于每个与三维平面相关联的像素 p ，在 p 的邻域中存在一个种子像素 q ，该像素也与 p 的平面相关联。总的来说，也可能没有或者有多个种子像素 q 。

$$\mathbf{C}_s(\mathbf{p}) = \mathbf{C}(\mathbf{p} + \mathbf{o}(\mathbf{p})) \quad D_s(u, v) = h(\mathbf{C}_s(u, v), u, v)$$

$$\mathbf{p} + \mathbf{o}(\mathbf{p}) + \mathbf{o}(\mathbf{p} + \mathbf{o}(\mathbf{p}))$$

$$D_f(u, v) = F(u, v)D_s(u, v) + (1 - F(u, v))D_i(u, v).$$

$$\mathcal{L}_{\text{depth}} = \mathcal{L}(D_f, D^*) + \lambda \mathcal{L}(D_s, D^*) + \mu \mathcal{L}(D_i, D^*)$$

6

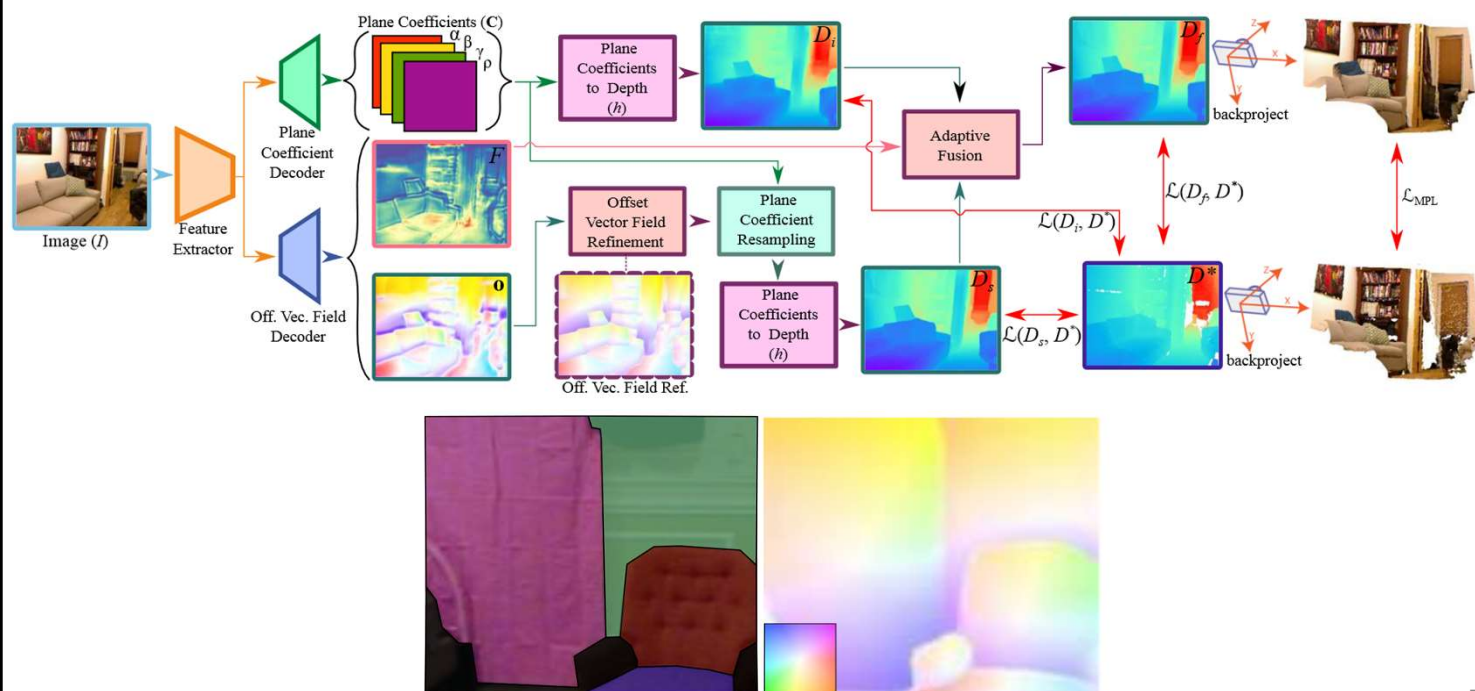
然而，先验并不总是有效，因此与基于种子的预测 D_s 相比，初始深度预测 D_i 实际上可能更好。 $F(u, v) \in [0, 1]$ ，表示模型使用预测种子像素通过 D_s 估计深度的置信度。

级联：同一平面区域内的种子像素应向区域中心靠拢，这有助于在预测区域平面系数时积累更多像素的信息

事实上，它可以简单地预测各处的零偏移量，并仍然产生有效的预测 D_s 和 D_f ，这将与 D_i 相同。

在实际应用中，由于神经网络映射 $f(\text{Image to Depth})$ 的规则性，初始预测 D_i 在深度边界附近被错误地平滑化，从而避免了这种不想要的行为。

因此，对于边界两侧的像素，预测一个远离边界的非零偏移值，会得到一个较低的Loss，因为这种偏移使用的 D_s 种子像素离边界更远，由于平滑化而产生的误差更小。



下面图展示了预测出的offset vector，大致意思是左下角不同的颜色代表offset不同的方向。蓝色表示应该向左上角偏移。

$\mathbf{A}\mathbf{n} = \mathbf{b}$, s.t. $\|\mathbf{n}\|_2 = 1$, where \mathbf{A} is a data matrix build by stacking the 3D points in the patch and \mathbf{b} is a vector of ones. Following [11, 53], the closed-form solution of this

$$\mathbf{n} = \frac{(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}}{\left\| (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \right\|_2}. \quad \mathcal{L}_{\text{MPL}} = \sum_{k=1}^K \|\mathbf{n}_k - \mathbf{n}_k^*\|_1.$$

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{depth}} + \mathcal{L}_{\text{MPL}}$$

法线是跨patch聚合，确保预测的表面和真实的表面的一阶一致性

Method	A.Rel	Log10	RMSE	δ_1	δ_2	δ_3
	Lower is better			Higher is better		
Plane detection based methods						
PlaneNet [42]	0.142	0.060	0.514	0.812	0.957	0.989
PlaneRCNN [41]	0.124	0.077	0.644	—	—	—
Yu <i>et al.</i> [81]	0.134	0.057	0.503	0.827	0.963	0.990
P ² Net (5F)* [80]	0.147	0.062	0.553	0.801	0.951	0.987
StruMonoNet [75]	0.107	0.046	0.392	0.887	0.980	0.995
Other monocular depth estimation methods						
Saxena <i>et al.</i> [59]	0.349	-	1.214	0.447	0.745	0.897
Karsch <i>et al.</i> [24]	0.349	0.131	1.21	-	-	-
Liu <i>et al.</i> [45]	0.335	0.127	1.06	-	-	-
Ladicky <i>et al.</i> [31]	-	-	-	0.542	0.829	0.941
Li <i>et al.</i> [37]	0.232	0.094	0.821	0.621	0.886	0.968
Wang <i>et al.</i> [68]	0.220	0.094	0.745	0.605	0.890	0.970
Liu <i>et al.</i> [44]	0.213	0.087	0.759	0.650	0.906	0.974
Roy <i>et al.</i> [57]	0.187	0.078	0.744	-	-	-
AdaBins [†] [1]	0.178	0.078	0.595	0.698	0.937	0.988
Eigen <i>et al.</i> [9]	0.158	-	0.641	0.769	0.950	0.988
Chakrabarti [4]	0.149	-	0.620	0.806	0.958	0.987
Li <i>et al.</i> [38]	0.143	0.063	0.635	0.788	0.958	0.991
Laina <i>et al.</i> [32]	0.127	0.055	0.573	0.811	0.953	0.988
Fu <i>et al.</i> [13]	0.115	0.051	0.509	0.828	0.965	0.992
Yin <i>et al.</i> [77]	0.108	0.048	0.416	0.875	0.976	0.994
Huynh <i>et al.</i> [23]	0.108	-	0.412	0.882	0.980	0.996
Lee <i>et al.</i> [33]	0.110	0.047	0.392	0.885	0.978	0.994
Long <i>et al.</i> [46]	0.101	<u>0.044</u>	0.377	0.890	<u>0.982</u>	<u>0.996</u>
Ranftl <i>et al.</i> [55]	0.110	0.045	<u>0.357</u>	0.904	0.988	0.998
Ours	<u>0.104</u>	0.043	0.356	0.898	0.981	0.996

Method	A.Rel	S.Rel	RMSE	RMSElog	δ_1	δ_2	δ_3
Lower is better					Higher is better		
Garg split [16] cap: 50m							
Garg <i>et al.</i> [16]	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Godard <i>et al.</i> [18]	0.108	0.657	3.729	0.194	0.873	0.954	0.979
Kuznetsov [29]	0.108	0.595	3.518	0.179	0.875	0.964	0.988
Gan <i>et al.</i> [15]	0.094	0.552	3.133	0.165	0.898	0.967	0.986
Fu <i>et al.</i> [13]	0.071	0.268	2.271	0.116	0.936	0.985	0.995
AdaBins [1]	0.058	0.19	2.36	0.088	<u>0.964</u>	<u>0.995</u>	0.999
Lee <i>et al.</i> [33]	<u>0.056</u>	<u>0.169</u>	<u>1.925</u>	<u>0.087</u>	<u>0.964</u>	0.994	0.999
Ours	0.055	0.130	1.651	0.081	0.974	0.997	0.999
Eigen split [9] cap: 80m							
Saxena <i>et al.</i> [59]	0.280	3.012	8.734	0.361	0.601	0.820	0.926
Eigen <i>et al.</i> [9]	0.203	1.548	6.307	0.282	0.702	0.898	0.967
Liu <i>et al.</i> [43]	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Godard <i>et al.</i> [18]	0.114	0.898	4.935	0.206	0.861	0.949	0.976
Kuznetsov [29]	0.113	0.741	4.621	0.189	0.862	0.960	0.986
Gan <i>et al.</i> [15]	0.098	0.666	3.933	0.173	0.890	0.964	0.985
Fu <i>et al.</i> [13]	0.072	0.307	<u>2.727</u>	0.120	0.932	0.984	0.994
Yin <i>et al.</i> [77]	0.072	-	3.258	0.117	0.938	0.990	0.998
Lee <i>et al.</i> [33]	0.059	0.245	2.756	<u>0.096</u>	<u>0.956</u>	<u>0.993</u>	<u>0.998</u>
AdaBins [1]	0.067	0.278	2.96	0.103	0.949	0.992	0.998
Ranftl <i>et al.</i> [55]	<u>0.062</u>	-	2.573	0.092	0.959	0.995	0.999
Ours	0.071	<u>0.270</u>	2.842	0.103	0.953	<u>0.993</u>	<u>0.998</u>

KITTI最远80米表现在不佳，远了找不到光滑的平面了。

0.055即使在2021也不是SOTA。不如MonoDELSNet，不过RMSE，和S.Rel确实很牛，比现在的SOTA（1.966）都要好。（可能异常值要少一点，因为平面）

Table 5. **Ablation study of components of our method.** *D*: directly predicting depth, *C*: predicting plane coefficients, “Guid.”: guidance module for plane coefficient decoder, “OV”: offset vectors, “Ref.”: cascaded refinement of offsets, “MPL”: mean plane loss, “+”: offset length is restricted to $\tau=0.3$ instead of $\tau=0.1$.

Pred.	Guid.	OV	Ref.	MPL	A.Rel ↓	RMSE ↓	δ_1 ↑
<i>D</i>					0.142	0.458	0.821
<i>C</i>					0.144	0.487	0.811
<i>C</i>	✓				0.142	0.458	0.824
<i>D</i>		✓			0.140	0.453	0.824
<i>C</i>		✓			0.116	0.390	0.877
<i>C</i>				✓	0.118	0.395	0.872
<i>C</i>	✓	✓			0.115	0.384	0.879
<i>C</i>	✓	✓+			0.116	0.390	0.879
<i>D</i>		✓	✓		0.134	0.440	0.839
<i>C</i>		✓	✓		0.113	0.378	0.884
<i>C</i>		✓	✓	✓	0.109	0.370	0.890
<i>C</i>	✓	✓	✓		0.109	0.373	0.889
<i>C</i>	✓	✓	✓	✓	0.104	0.356	0.898

我们观察到，在独立设置中，直接预测深度比预测平面系数更好。然而，一旦我们插入预测偏移向量的第二个头，与直接预测深度相比，使用平面系数表示可以获得显著的好处。这表明，由于平面系数表示，网络学会了有效利用种子像素处的局部平面信息来提高深度。此外，添加我们的指导模块提供了轻微的改进

（这个多加个Mean Plane Loss 效果那么好？）

Guidance大致基于[34]。每个解码器块的输出经过平面系数引导模块，生成 4 通道平面系数。

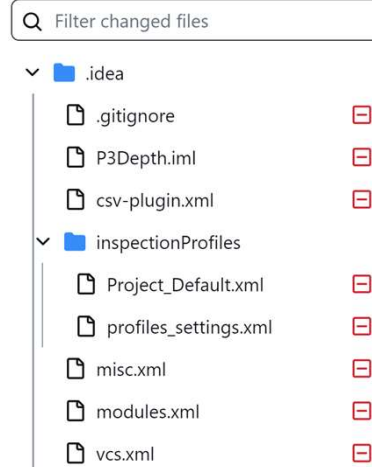
引导模块的输出大小被上采样以匹配最后一个解码器层的输入大小。最后，这些来自每个尺度的平面系数被转换为深度。所有这些深度图都通过的前一个decoder传递到最后一个decoder的feature map连接起来。

Talk is cheap, show me your code!



[CVPR 2022 \(pdf\)](#) | [arXiv \(pdf\)](#) | [Project page](#)

This repository is still being updated !!!



类似的，数据集来源同样有问题，README中的链接指向另一个有问题的repo，补完后仍然缺少文件

Talk is cheap, show me your code!



After setting up the environment, I used NYU dataset for training, but the training results were very strange. The loss function converged slowly, rmse kept increasing, and delta kept decreasing.

BayMaxBHL commented on Feb 16

Author ...

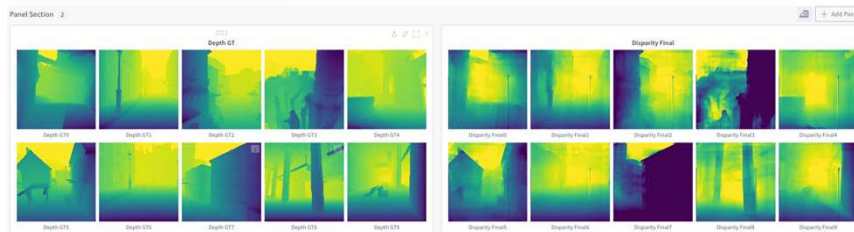
@haifengwu205 确实是用不了，我这不晒出来的结果就是不收敛嘛。rmse还卡卡往上涨，人都麻了。

@BayMaxBHL This is my code https://drive.google.com/file/d/1RRhOknM4tPnWzvi-T0B_BxLWE8famL3/view?usp=sharing.

I training with gta dataset. I add GTA_dataset.py to load color and depth image.

• More details on evaluation and pretrained models will be released soon.

After 40 epochs, the result is bad. I gave up.

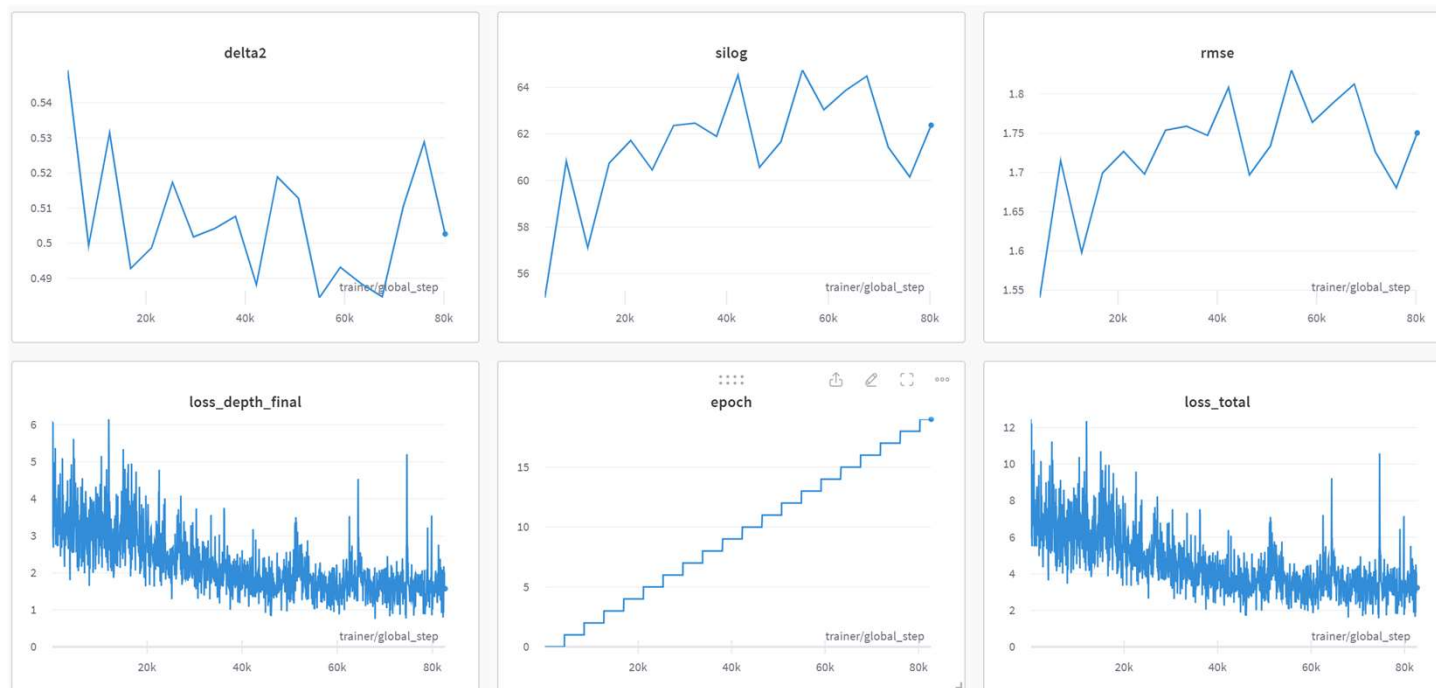


我们观察到，在独立设置中，直接预测深度比预测平面系数更好。然而，一旦我们插入预测偏移向量的第二个头，与直接预测深度相比，使用平面系数表示可以获得显著的好处。这表明，由于平面系数表示，网络学会了有效利用种子像素处的局部平面信息来提高深度。此外，添加我们的指导模块提供了轻微的改进

（这个多加个Mean Plane Loss 效果那么好？）

Guidance大致基于[34]。每个解码器块的输出经过平面系数引导模块，生成 4 通道平面系数。

引导模块的输出大小被上采样以匹配最后一个解码器层的输入大小。最后，这些来自每个尺度的平面系数被转换为深度。所有这些深度图都通过的前一个decoder传递到最后一个decoder的feature map连接起来。



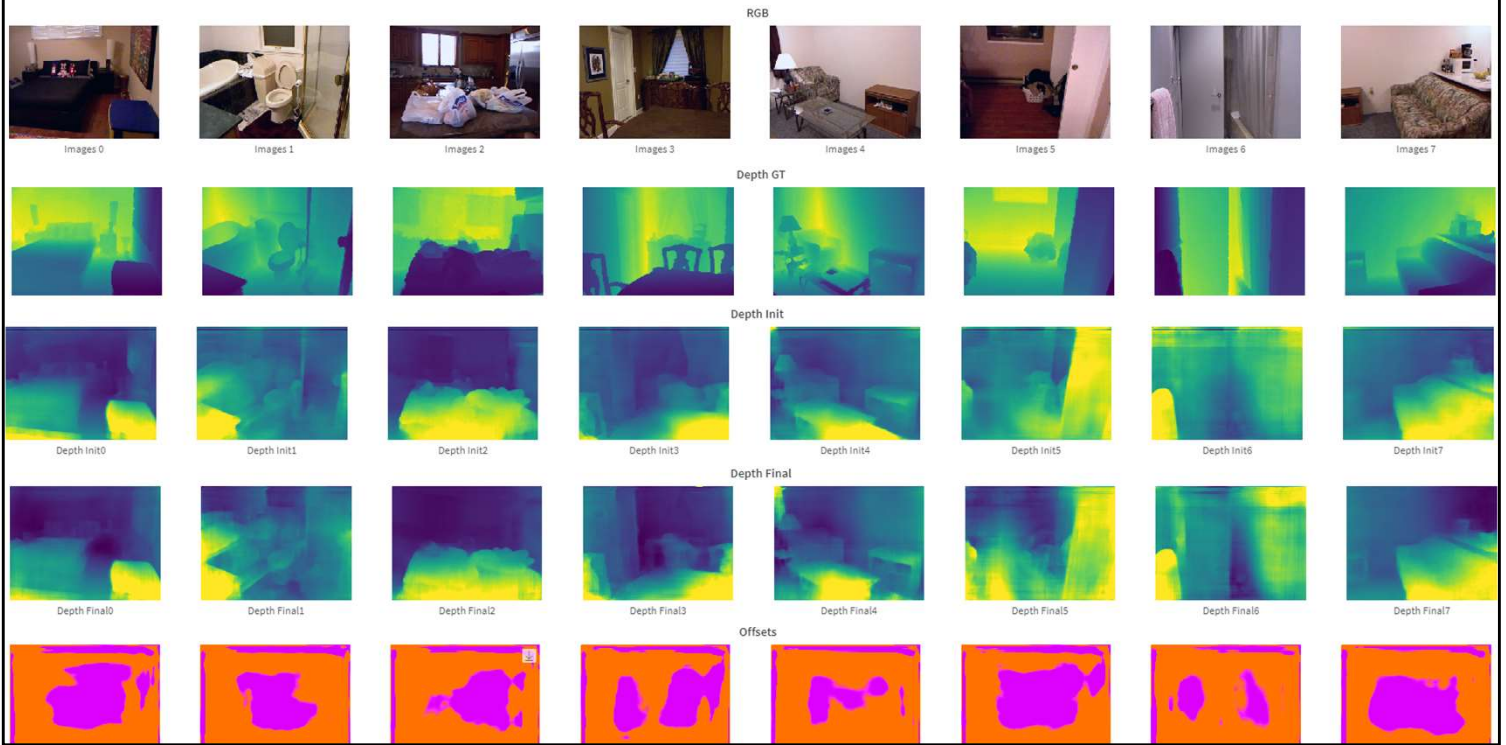
我们观察到，在独立设置中，直接预测深度比预测平面系数更好。然而，一旦我们插入预测偏移向量的第二个头，与直接预测深度相比，使用平面系数表示可以获得显着的好处。这表明，由于平面系数表示，网络学会了有效利用种子像素处的局部平面信息来提高深度。此外，添加我们的指导模块提供了轻微的改进

（这个多加个Mean Plane Loss 效果那么好？）

Guidance大致基于[34]。每个解码器块的输出经过平面系数引导模块，生成 4 通道平面系数。

引导模块的输出大小被上采样以匹配最后一个解码器层的输入大小。最后，这些来自每个尺度的平面系数被转换为深度。所有这些深度图都通过的前一个decoder传递到最后一个decoder的feature map连接起来。

Talk is cheap, show me your result!





Thanks