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

曾德御

2023.08.17

Contribution

① **Learning to Identify Seed Pixels**

② **Mean Plane Loss.**

2

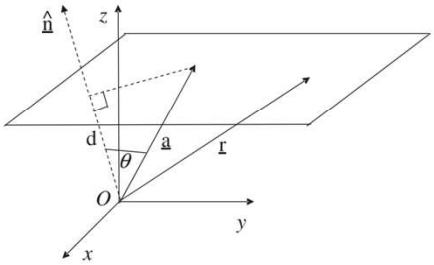
Depth Plane的CoEfficient借鉴于GeoLayOut.
Offset Vector借鉴于Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth.

Prerequisites: Cartesian Representation of Plane

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

5.1.1 Plane from vector to Cartesian form

- $(\mathbf{r}-\mathbf{a}).\hat{\mathbf{n}} = 0$ gives $\mathbf{r}.\hat{\mathbf{n}} = \mathbf{a}.\hat{\mathbf{n}}$
- Note that $d = a\cos\theta = \mathbf{a}.\hat{\mathbf{n}}$ is the perpendicular distance of the plane to the origin.
- Also we write $\hat{\mathbf{n}} = l\mathbf{i} + m\mathbf{j} + n\mathbf{k}$. where (l, m, n) are defined as the *direction cosines* of the normal to the plane.



- Finally we write the general vector $\mathbf{r}$ as (x, y, z)
- This gives the plane in Cartesian representation as

$$\mathbf{r}.\hat{\mathbf{n}} = lx + my + nz = d$$

3

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Poisitional 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。

Method: Plane Coefficient Representation for Depth

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

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

$\rho = \sqrt{\hat{\alpha}^2 + \hat{\beta}^2 + \hat{\gamma}^2}$ and normalizing $\alpha = \frac{\hat{\alpha}}{\rho}$, $\beta = \frac{\hat{\beta}}{\rho}$ and $\gamma = \frac{\hat{\gamma}}{\rho}$ 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Poisitional Vector的意思。

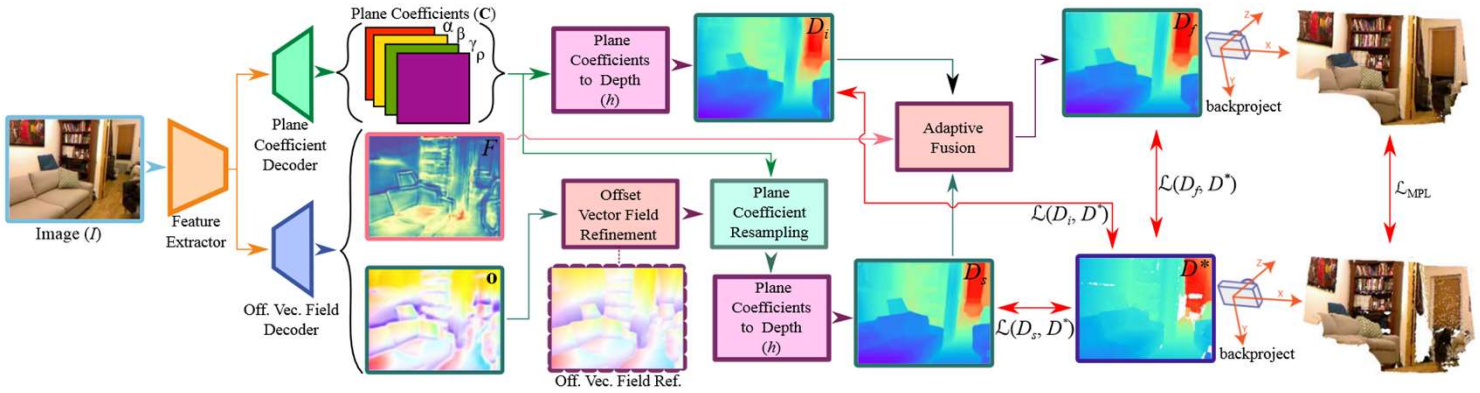
Cartesian representation

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。

Network: Plane Coefficient Representation for Depth

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

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



5

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 p，在 p 的邻域中存在一个种子像素 q，该种子像素 q 也与与 p 相同的平面关联。总的，p 可能存在多个种子像素或没有种子像素

Network：Learning to Indentify seed Seed Pixels

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。

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

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

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

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

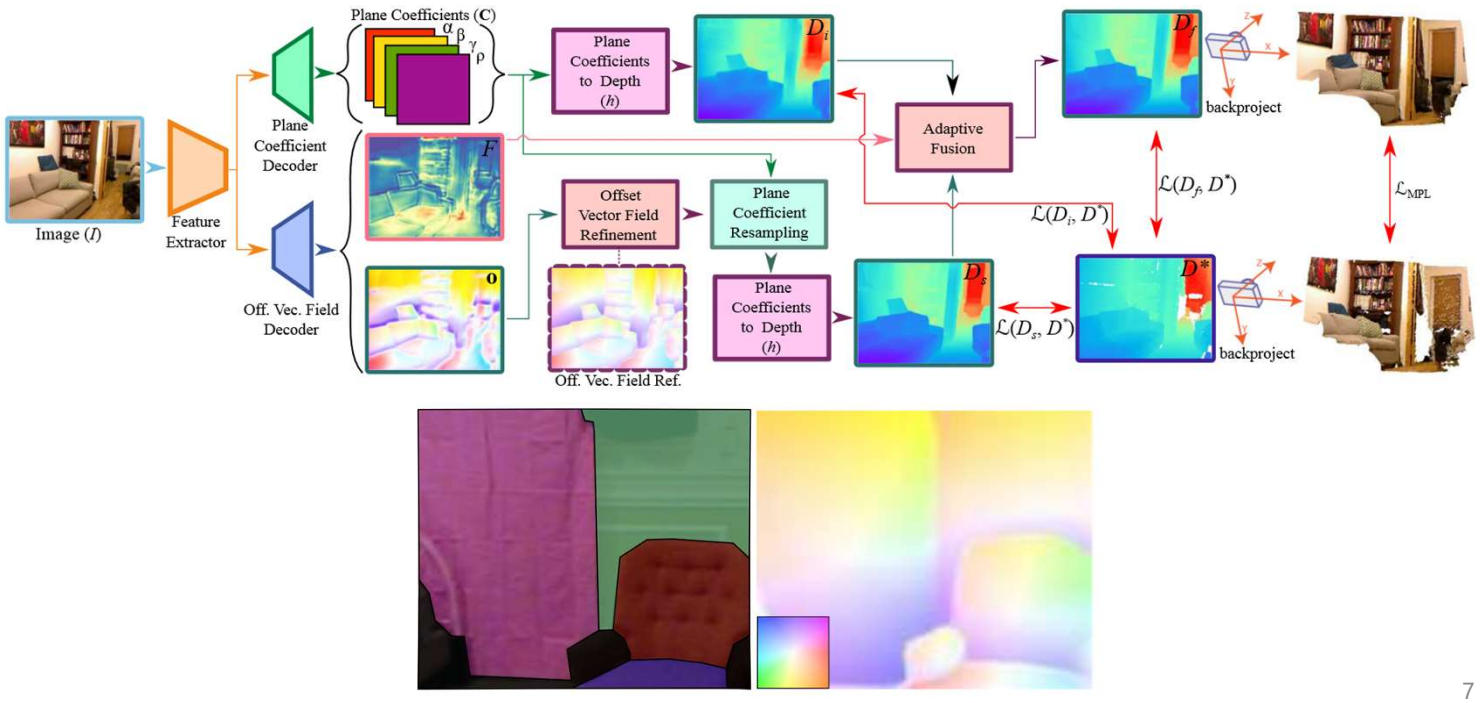
6

然而，先验并不总是有效，因此与基于种子的预测 Ds 相比，初始深度预测 Di 实际上可能更好。F (u, v) ∈ [0, 1]，表示模型使用预测种子像素通过 Ds 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Ds和Df，这将与Di相同。
在实际应用中，由于神经网络映射 f(Image to Depth)的规则性，初始预测 Di 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Ds种子像素离边界更远，由于平滑化而产生的误差更小。

Network: Overview



7

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。

Network： Mean Plane Loss

$\mathbf{An} = \mathbf{b}$, 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

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

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

8

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

Experiment

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 *et al.* [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 *et al.* [59]	0.349	-	1.214	0.447	0.745	0.897
Karsch *et al.* [24]	0.349	0.131	1.21	-	-	-
Liu *et al.* [45]	0.335	0.127	1.06	-	-	-
Ladicky *et al.* [31]	-	-	-	0.542	0.829	0.941
Li *et al.* [37]	0.232	0.094	0.821	0.621	0.886	0.968
Wang *et al.* [68]	0.220	0.094	0.745	0.605	0.890	0.970
Liu *et al.* [44]	0.213	0.087	0.759	0.650	0.906	0.974
Roy *et al.* [57]	0.187	0.078	0.744	-	-	-
AdaBins† [1]	0.178	0.078	0.595	0.698	0.937	0.988
Eigen *et al.* [9]	0.158	-	0.641	0.769	0.950	0.988
Chakrabarti [4]	0.149	-	0.620	0.806	0.958	0.987
Li *et al.* [38]	0.143	0.063	0.635	0.788	0.958	0.991
Laina *et al.* [32]	0.127	0.055	0.573	0.811	0.953	0.988
Fu *et al.* [13]	0.115	0.051	0.509	0.828	0.965	0.992
Yin *et al.* [77]	0.108	0.048	0.416	0.875	0.976	0.994
Huynh *et al.* [23]	0.108	-	0.412	0.882	0.980	0.996
Lee *et al.* [33]	0.110	0.047	0.392	0.885	0.978	0.994
Long *et al.* [46]	**0.101**	0.044	0.377	0.890	0.982	0.996
Ranftl *et al.* [55]	0.110	0.045	0.357	**0.904**	**0.988**	**0.998**
Ours	0.104	**0.043**	**0.356**	0.898	0.981	0.996

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

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）

Ablation

Table 5. **Ablation study of components of our method.** D: directly predicting depth, $\mathbf{C}$: 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 τ=0.3 instead of τ=0.1.

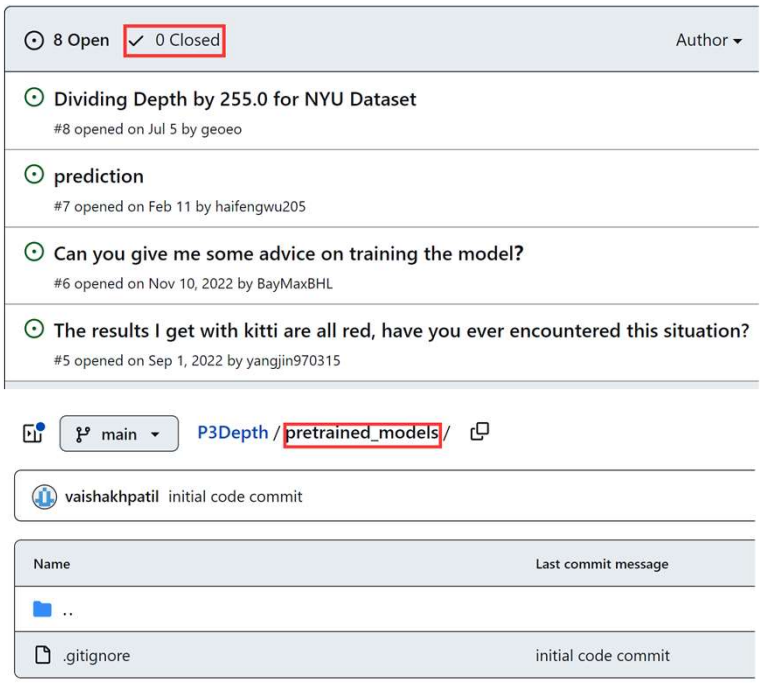
Pred.	Guid.	OV	Ref.	MPL	A.Rel ↓	RMSE ↓	δ_1 ↑
D					0.142	0.458	0.821
$\mathbf{C}$					0.144	0.487	0.811
$\mathbf{C}$	✓				0.142	0.458	0.824
D		✓			0.140	0.453	0.824
$\mathbf{C}$		✓			0.116	0.390	0.877
$\mathbf{C}$				✓	0.118	0.395	0.872
$\mathbf{C}$	✓	✓			0.115	0.384	0.879
$\mathbf{C}$	✓	✓+			0.116	0.390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

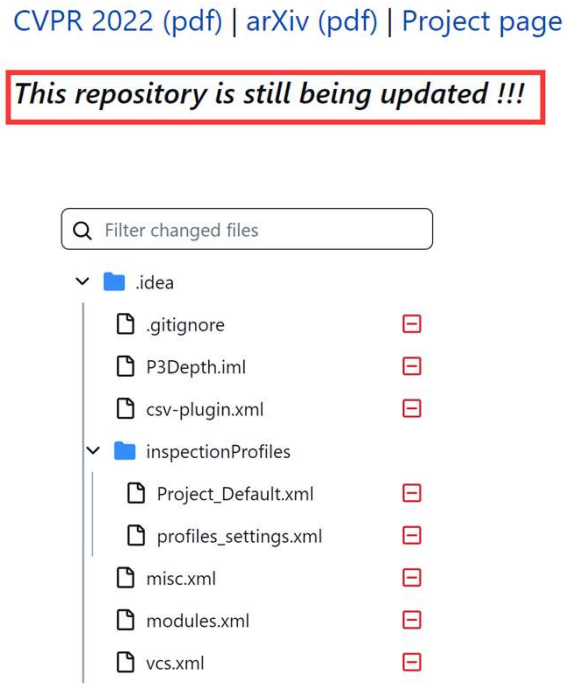
我们观察到，在独立设置中，直接预测深度比预测平面系数更好。然而，一旦我们插入预测偏移向量的第二个头，与直接预测深度相比，使用平面系数表示可以获得显着的好处。这表明，由于平面系数表示，网络学会了有效利用种子像素处的局部平面信息来提高深度。此外，添加我们的指导模块提供了轻微的改进

（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 code!





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!

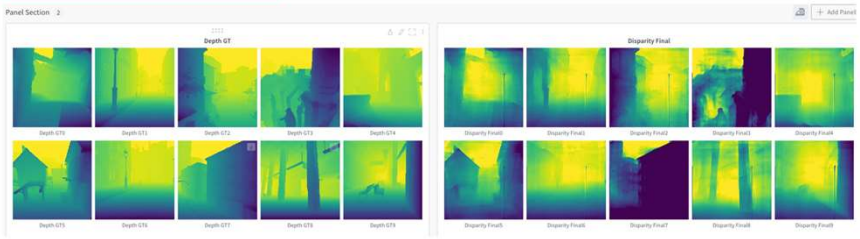
After setting up the environment, I used NYU dataset for training, but the training results were very strange.
The loss function converged slowly, rmse kept increasing, and delta kept decreasing.

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

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

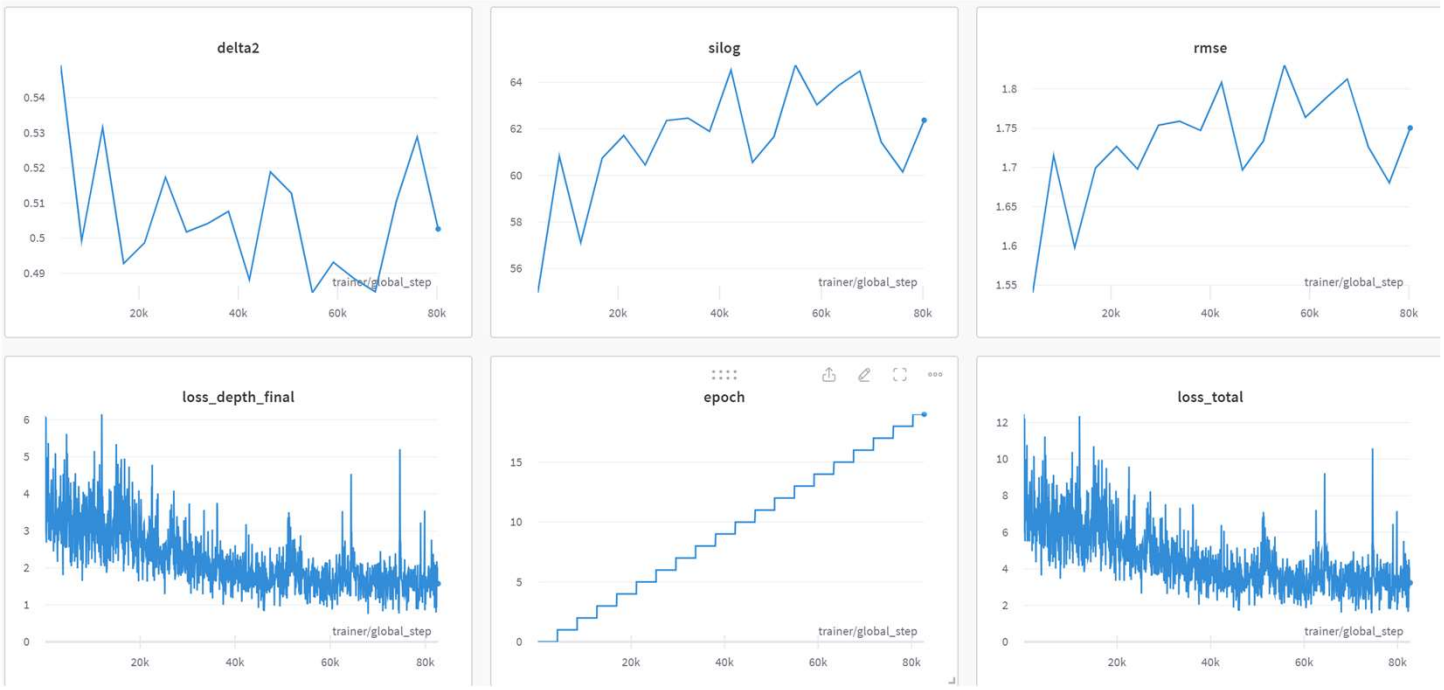


我们观察到，在独立设置中，直接预测深度比预测平面系数更好。然而，一旦我们插入预测偏移向量的第二个头，与直接预测深度相比，使用平面系数表示可以获得显着的好处。这表明，由于平面系数表示，网络学会了有效利用种子像素处的局部平面信息来提高深度。此外，添加我们的指导模块提供了轻微的改进

（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!

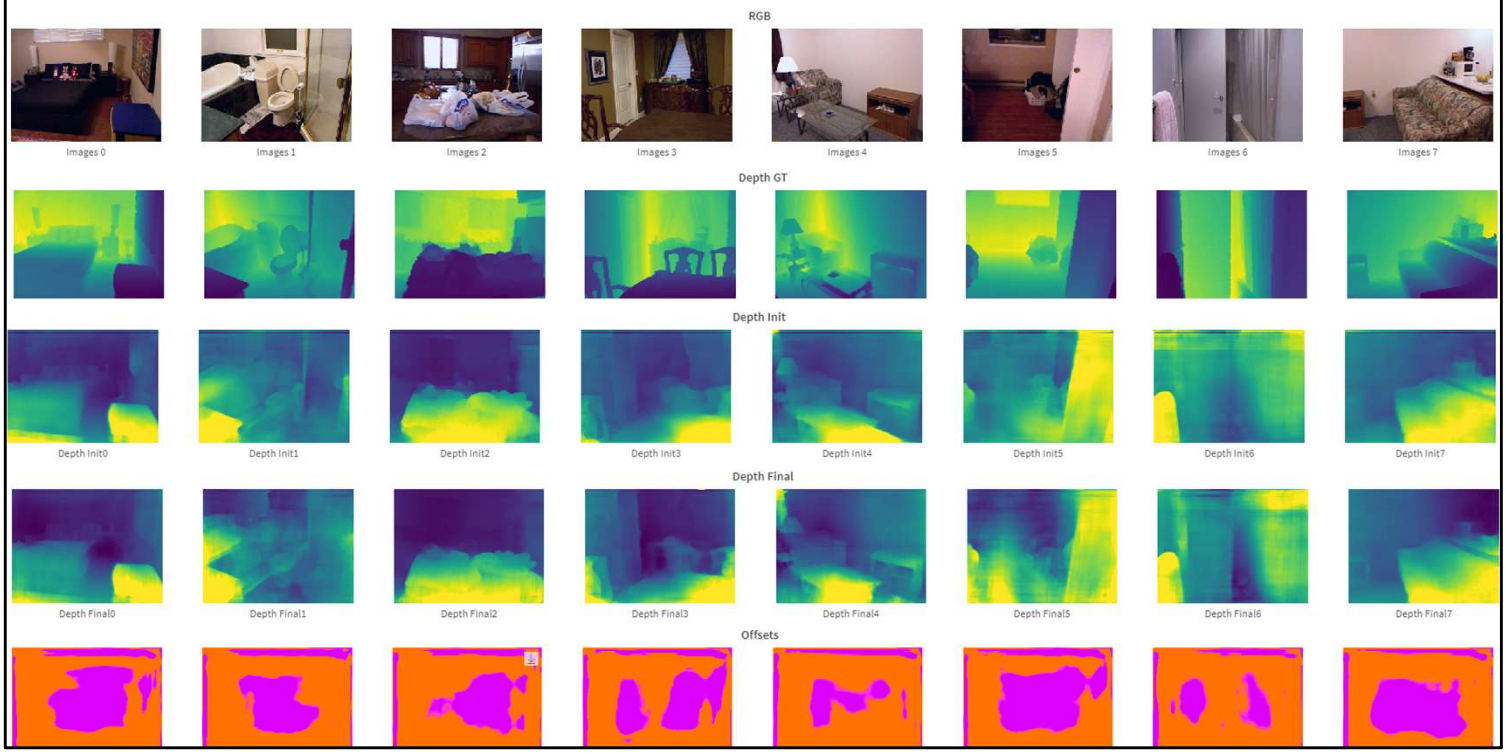


我们观察到，在独立设置中，直接预测深度比预测平面系数更好。然而，一旦我们插入预测偏移向量的第二个头，与直接预测深度相比，使用平面系数表示可以获得显着的好处。这表明，由于平面系数表示，网络学会了有效利用种子像素处的局部平面信息来提高深度。此外，添加我们的指导模块提供了轻微的改进

（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!



DIG

Thanks