DIG

DualRefine: Self-Supervised Depth and Pose Estimation Through Iterative Epipolar Sampling and Refinement Toward Equilibrium

2023.07.20

Contribution & Motivation

① **Iterative update module.** (DualRefine: 1. refine the depth; 2. refine the pose network.)

② Refine updates only on local cost volumes, **efficent and robust.**

③ **DEQ framework with less memory consumption**

① Dro: Deep recurrent optimizer for structure-from-motion (RAL 2021)

② Deep Equilibrium Models (NeurIPS 2019)

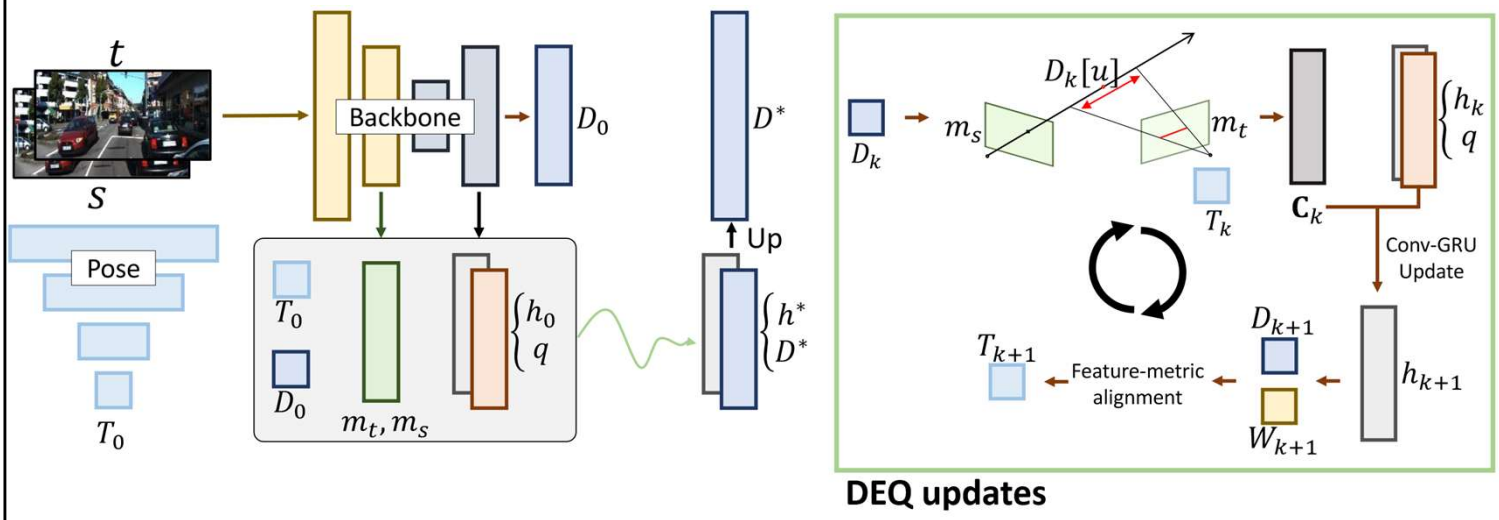
③ Deep equilibrium optical flow estimation (CVPR 2022)

④ Raft: Recurrent all-pairs field transforms for optical flow (ECCV 2020 Best Paper)

2

第一点借鉴了1 3 4

Network: Overview



backbone是Diffnet（HRNet作为decoder，比ResNet效果好），应该是排名第四的网络。当时的sota。

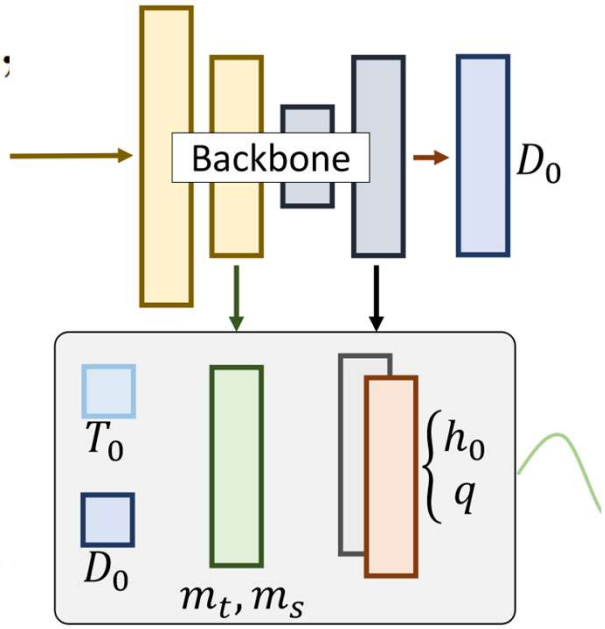
Network: Deep equilibrium alignments

$$(h^*, D^*, T^*) = z^* = \mathrm{U}(z^*, x),$$

$$h^{[0]} = \tanh(\mathrm{H}(F^{(1/4)}))$$

$$q = \mathrm{Q}(F^{(1/4)})$$

$m_s^{(1/4)}$ and the target image $m_t^{(1/4)}$



update的是 hidden state、depth、pose, 在scale s = 2 下迭代执行。backbone是Diffnet（HRNet作为decoder，比ResNet效果好），应该是排名第四的网络。当时的sota。
D_0, T_0是DiffNet（backbone）给出的。

Network: Depth Updates Around Local Neighborhood

$$D[u]_k \pm (i \times c \times n)$$

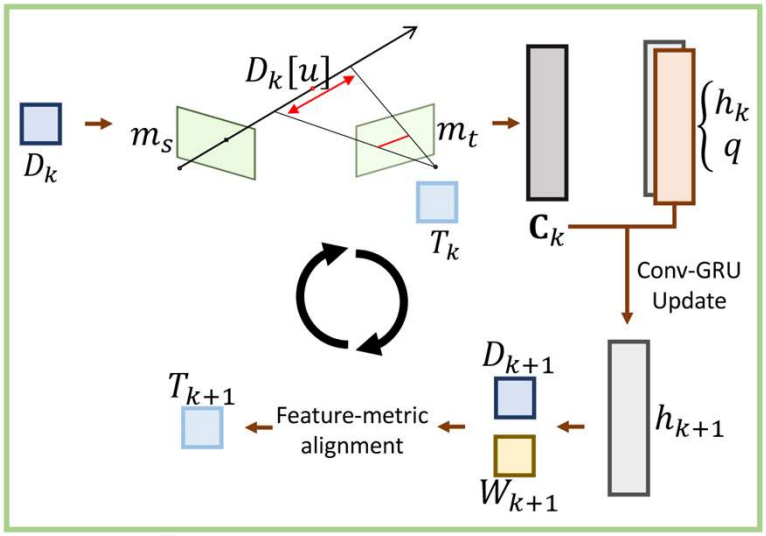
$i \in \mathbb{Z}$ and $i \leq r \quad c = D[u]/C$

set C as a trainable parameter.

$$n = \{1, 2, 3\}$$

$$\mathbf{C}_k[u] = |m_t[u] - m_s\langle u'_k \rangle|$$

$$x_k = [\mathrm{CNN}_{\mathbf{C}}(\mathbf{C}_k), \mathrm{CNN}_{D_x}(D_k), q],$$



DEQ updates

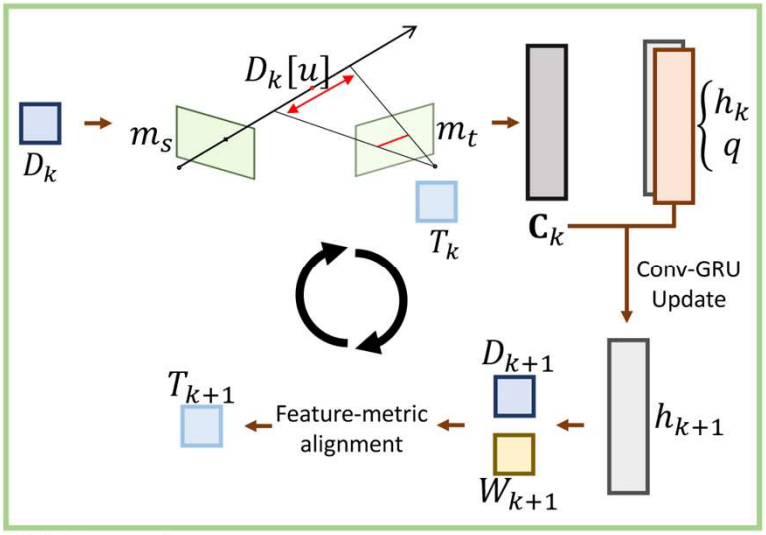
5

不知道图片的notation为什么和公式上的不一致，左边应该是mt, 右边是m_s。
r = 8，在深度估计中，误差通常随距离的增加而增加。为了考虑到这一点，我们将 c = D[u]/C 定义为深度的函数，使采样范围取决于深度。
在每一层，我们以1/2^n的比例对源图像m_s^(1/4)的匹配特征图进行双线性resize。(意思是偏离越远m_s下采样越小)然后，在计算出的相应坐标集u'k上对匹配特征进行采样，并计算与目标特征的绝对差值。
Ck_u就是由 n * (2 * r + 1)个区别构成的。
然后把这些过一次两层的CNN再和 q 连接（concatenate）起来

Network: Depth Updates Around Local Neighborhood

$$z_{k+1} = \sigma(\mathrm{CNN}_z([h_k, x_k]))$$
$$r_{k+1} = \sigma(\mathrm{CNN}_r([h_k, x_k]))$$
$$\tilde{h}_{k+1} = \tanh(\mathrm{CNN}_{\tilde{h}}([r_{k+1} \odot h_k, x_k]))$$
$$h_{k+1} = (1 - z_{k+1}) \odot h_k + z_{k+1} \odot \tilde{h}_{k+1}$$

$$D_{k+1} = D_k + r \cdot c \cdot \tanh(\mathrm{CNN}_{D_U}(h_{k+1})).$$



DEQ updates

Network: Feature-metric pose alignments

$$H_k = \mathcal{J}_k^T diag(W_k)\mathcal{J}_k \text{ and } b_k = -\mathcal{J}_k^T diag(W_k) r_k$$

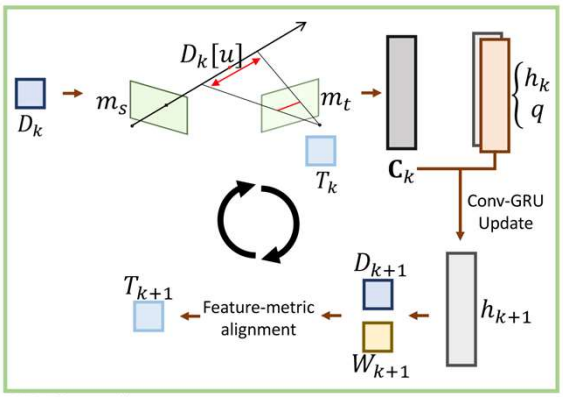
$$r_k[u] = m_s\langle u'_k \rangle - m_t[u],$$

$$H_k \delta_k = b_k$$

$$W_{t,s} = 1/(1 + \text{ReLU}(\text{CNN}_{W_q}(F_{t,s})))$$

$$W_q = W_t \cdot W_{s,warped} \quad W_{h,k} = \text{CNN}_{W_h}(h_k)$$

$$T_{k+1} = exp(\delta_k^{\wedge}) T_k$$



DEQ updates

7

因此，Hδ = b 表示一个线性方程组，其中 H 矩阵表示特征点或描述子之间的相似性，δ 向量表示图像之间的变换，b 向量表示特征点或描述子之间的差异。通过解决这个线性方程组，可以实现直接特征对齐，将图像特征点或描述子对齐到共同的参考坐标系中。

J is the Jacobian with respect to the pose

这里的W是置信度，

由于pose会被移动物体和重复纹理影响，可以对输入上下文特征图的置信度加权：Wq

pose又会被深度估计的精度影响，精度高的区域要分配更高的置信度Wh,k，由于隐藏状态h具有这些匹配成本的历史

在我们的实验中，我们研究了每种置信度的使用以及两者的组合Wk = WqWh,k。

Network: Deq Training

在不动点上，我们为每个refine的深度D*和姿态T*估计值计算两个(?)额外的自监督损失。之前的工作是将多帧深度估计值D*与教师姿势估计值T0配对，以执行warp损失计算，而我们则将D*与T*配对。

	Loss pairs D^*	Loss pairs T^*	Abs Rel ↓	Sq Rel ↓	RMSE ↓	δ_1 ↑
1	T_0	D_0	0.99	0.765	4.449	0.898
2	T^*	D_0	0.093	0.698	4.342	0.907
3	T_0	D^*	0.092	0.657	4.34	0.908
4	T^*	D^*	0.089	0.632	4.305	0.907

Network: Deq Training

借鉴了ManyDepth中的consistency loss (D0 and D∗之间的)

$$L_{\text{consistency}} = \sum M\,|D_t - \hat{D}_t|. \qquad M = \max\left(\frac{D_{\text{cv}} - \hat{D}_t}{\hat{D}_t}, \frac{\hat{D}_t - D_{\text{cv}}}{D_{\text{cv}}}\right) > 1.$$

$$L = (1 - M)L_p + L_{\text{consistency}}$$

与ManyDepth不同的是，并不明确构建Cost Volume。为了获得粗深度（对应D_{cv}），我们搜索教师深度邻域附近的最低匹配成本，但邻域范围更大。与基于代价体积的方法相比，这种方法具有额外的优势，因为我们不需要依赖估计的最小深度和最大深度，也不需要知道估计值的范围。此外，由于粗深度的计算取决于匹配成本的准确性，因此可以通过更准确的姿态估计来改进粗深度的计算。

9

具体来说，我们从原始特征匹配中提取粗略的深度预测值，并掩盖出现较大分歧的区域，同时强化与教师深度预测值的一致性。
（不可信就不要乱优化了，但是ManyDepth使用的是单帧网络(in general they make far less severe mistakes on moving objects.)）

Experiment

	Method	Test frames	Semantics	$W \times H$	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓	δ_1 ↑	δ_2 ↑	δ_3 ↑
Low & mid res	Ranjan *et al.* [73]	1		832 × 256	0.148	1.149	5.464	0.226	0.815	0.935	0.973
	EPC++ [62]	1		832 × 256	0.141	1.029	5.350	0.216	0.816	0.941	0.976
	Struct2depth (M) [11]	1	•	416 × 128	0.141	1.026	5.291	0.215	0.816	0.945	0.979
	Videos in the wild [29]	1	•	416 × 128	0.128	0.959	5.230	0.212	0.845	0.947	0.976
	Guizilini *et al.* [33]	1	•	640 × 192	0.102	0.698	4.381	0.178	0.896	0.964	**0.984**
	Johnston *et al.* [45]	1		640 × 192	0.106	0.861	4.699	0.185	0.889	0.962	0.982
	Monodepth2 [26]	1		640 × 192	0.115	0.903	4.863	0.193	0.877	0.959	0.981
	Packnet-SFM [31]	1		640 × 192	0.111	0.785	4.601	0.189	0.878	0.960	0.982
	Li *et al.* [54]	1		416 × 128	0.130	0.950	5.138	0.209	0.843	0.948	0.978
	DIFFNet [109]	1		640 × 192	0.102	0.764	4.483	0.180	0.896	0.965	0.983
	DualRefine-initial (D_0)	1		640 × 192	0.103	0.721	4.476	0.180	0.891	0.965	**0.984**
	Patil *et al.* [70]	N†		640 × 192	0.111	0.821	4.650	0.187	0.883	0.961	0.982
	Wang *et al.* [93]	2 (-1, 0)		640 × 192	0.106	0.799	4.662	0.187	0.889	0.961	0.982
	ManyDepth (MR) [95]	2 (-1, 0)		640 × 192	0.098	0.770	4.459	0.176	0.900	0.965	0.983
	DepthFormer [32]	2 (-1, 0)		640 × 192	0.090	**0.661**	**4.149**	0.175	0.905	**0.967**	**0.984**
	DualRefine-refined (D^*)	2 (-1, 0)		640 × 192	**0.087**	0.698	4.234	**0.170**	**0.914**	**0.967**	0.983
High res	DRO [30]	2 (-1, 0)		960 × 320	0.088	0.797	4.464	0.212	0.899	0.959	0.980
	Wang *et al.* [93]	2 (-1, 0)		1024 × 320	0.106	0.773	4.491	0.185	0.890	0.962	0.982
	ManyDepth (HR ResNet50) [95]	2 (-1, 0)		1024 × 320	0.091	0.694	4.245	0.171	0.911	0.968	0.983
	DualRefine-refined (HR) (D^*)	2 (-1, 0)		960 × 288	**0.087**	**0.674**	**4.130**	**0.167**	**0.915**	**0.969**	**0.984**

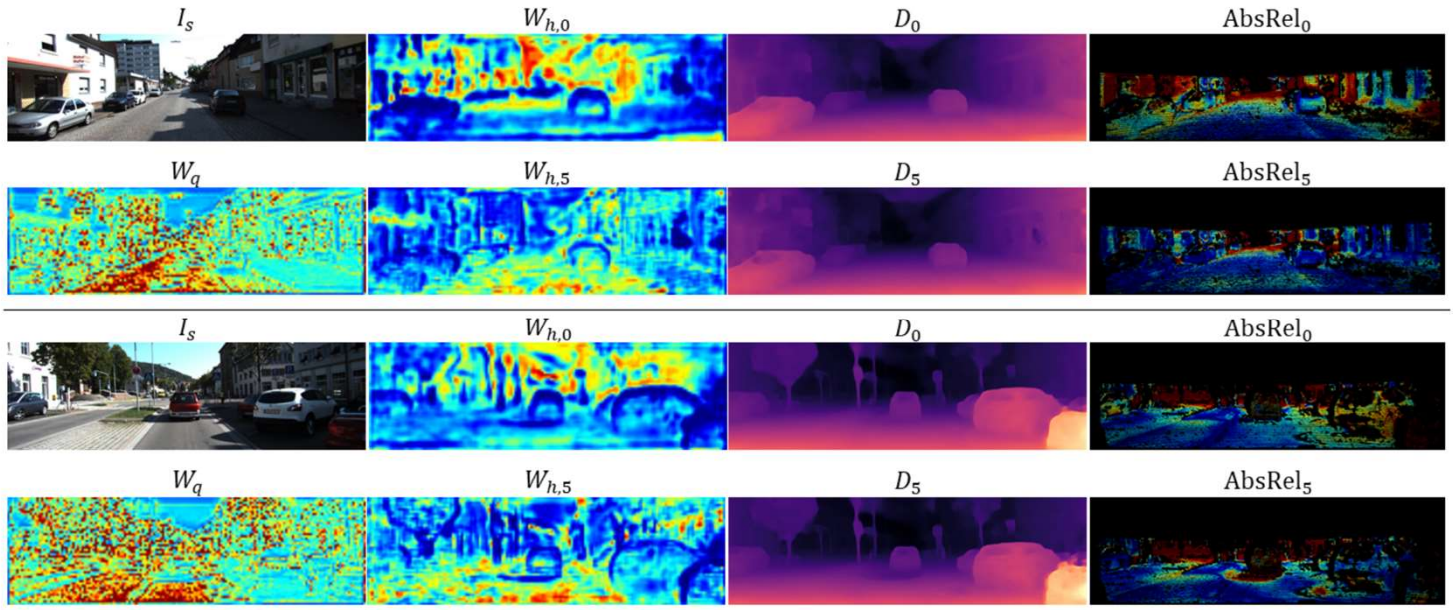
Ablation

Pose Updates	Consistency mask	Abs Rel	Sq Rel	RMSE	δ_1	δ_2
no update	T_0	0.097	0.713	4.462	0.898	0.964
no weights	T_0	0.091	0.694	4.271	0.909	**0.967**
no $W_{h,k}$	T_0	0.090	0.667	4.252	0.909	**0.967**
no W_q	T_0	0.093	0.686	4.258	0.908	**0.967**
W_q and $W_{h,k}$	T_0	0.090	0.669	4.293	0.910	**0.967**
no weights	T^*	0.092	0.667	4.257	0.908	**0.967**
no $W_{h,k}$	T^*	0.091	**0.666**	4.243	0.909	**0.967**
no W_q	T^*	0.088	0.674	4.251	0.911	0.966
W_q and $W_{h,k}$	T^*	**0.087**	0.698	**4.234**	**0.914**	**0.967**

DEQ # iters	Abs Rel	Sq Rel	RMSE	δ_1	Time (ms)
3→3	0.094	0.725	4.355	0.906	53
6→3	0.097	0.711	4.312	0.908	53
6→6	0.087	0.698	4.234	0.914	68
12→3	0.098	0.73	4.370	0.900	53
12→6	0.093	0.695	4.310	0.906	68
12→12	0.089	0.692	4.242	0.910	99

不过用位姿矩阵是什么意思呢？（应该是θconsistency shares θpose with our main network to help ensure scale-consistent predictions between θdepth and θconsistency）有趣的是，即使不使用权重来指导姿态计算，所有姿态更新模型也能获得相似的性能。额？

Depth Result



有趣的是，随着每次迭代而演变的置信度权重最初对较远的点赋予较高的置信度，并随着迭代次数的增加而向较近的点移动？

DIG

Thanks