A Comparison Between Three Optical Flow Estimate Method: FlowNet2, PWC-Net, GeoNet

Abstract

本文主要对比对象为在 CVPR2017 及 CVPR2018 上提出的三种端到端的、利用 CNN 进行光流估计的方法，分别为 FlowNet2[1]、PWC-Net[2]、GeoNet[3]。从三者的结构及其在 KITTI2012、KITTI2015、Sintel-Final 三个数据集上的性能这两个维度进行了对比，同时简要介绍了 FlowNet2 及 PWC-Net 在近年提出的衍生型 LiteFlowNet2 和 PWC-Net+。最终结果为 PWC-Net 在准确度、速度、计算资源上都具备一定优势，而 PWC-Net+及 LiteFlowNet2 则表现出更加先进的性能。

关键词：Optical Estimate, FlowNet2, PWC-Net, GeoNet

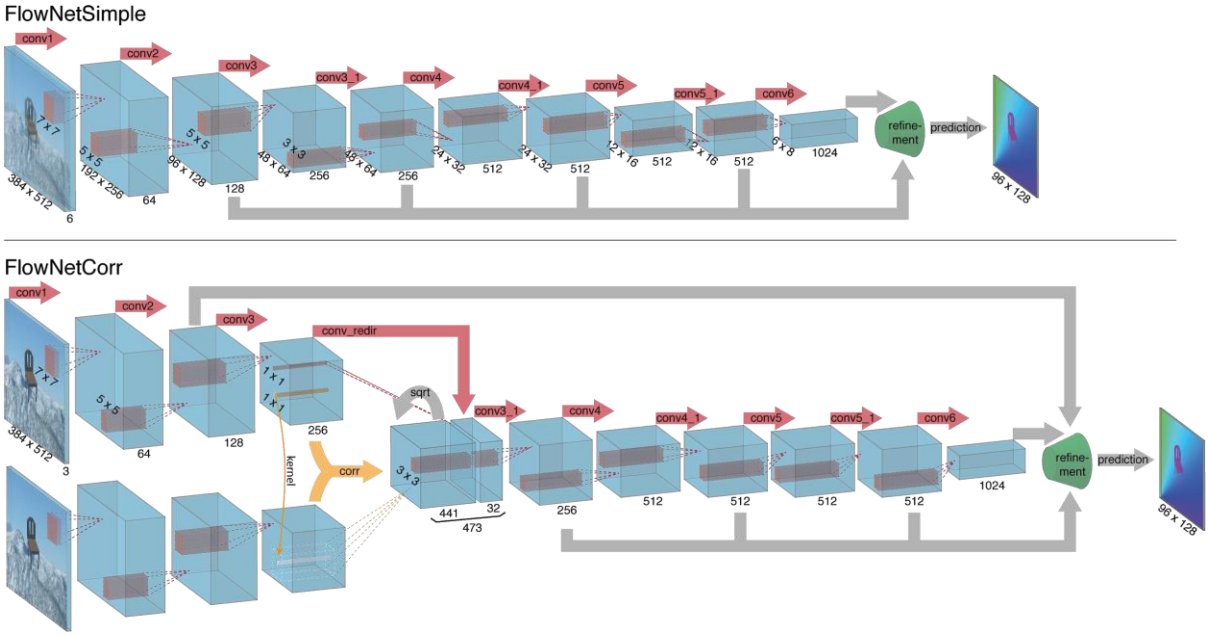
目录

1 FlowNet 2.0

1.1 Introduction:

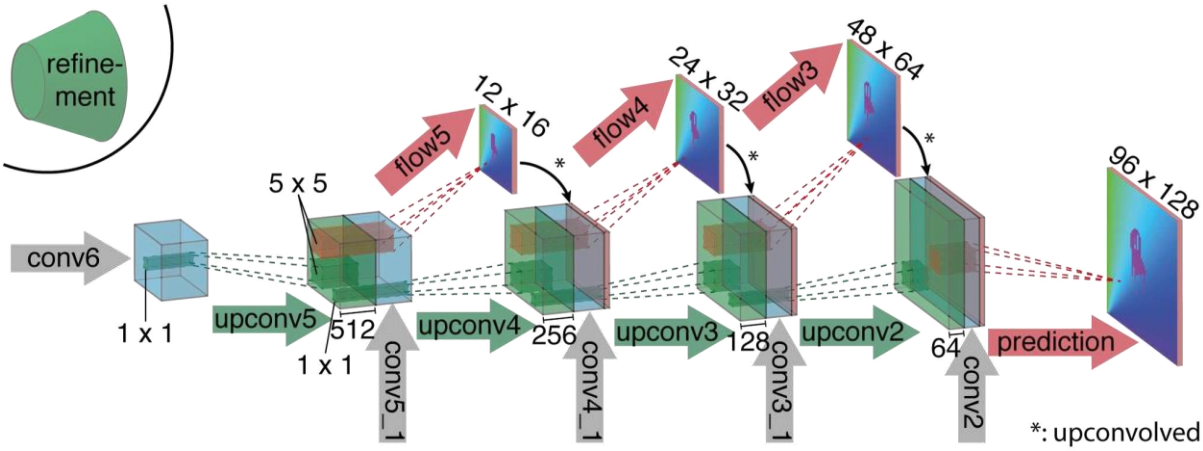
1.1.1 FlowNet:

FlowNet 是一种端到端的卷积神经网络（Convolutional Neutral Network, CNN）用来根据一对图片预测光流域（optical flow field）。由于不清楚一个标准的 CNN 结构是否能达到目标，作者既开发了一个标准的网络 FlowNetSimple，也开发了一个有关联层（correlation layer）的网络 FlowNetCorr。后者能够在多种规模下提取特征，从而根据特征找到两图的一致性来预测光流域。但作者发现带关联层的网络并没有在准确率上有明显的提升。



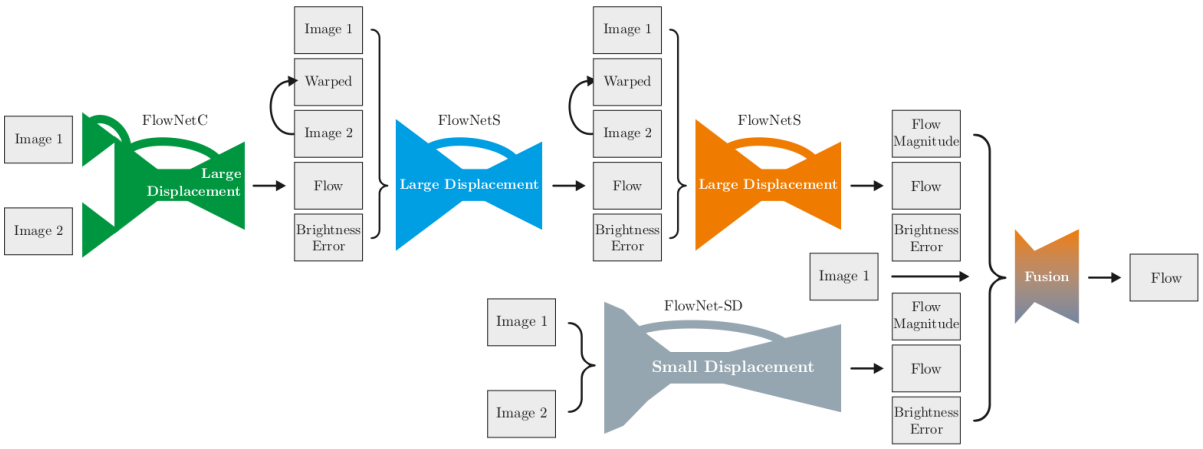
FlowNetSimple 直接将两个输入图片叠在一起，使用它来训练神经网络。训练过程中允许网络自身决定怎样处理输入的图片对来提取动作信息。FlowNetCorr 则是先对两个图片创造两个独立但相同的处理流（processing stream），对两个图片分别卷积得到有意义的特征图，再用关联层匹配它们的特征图，最后用关联层匹配的结果预测光流。两种方法最后都要经过优化，优化目的是还原分辨率得到较清晰的光流域。

关联层对两个特征图的每个图像块进行乘法对比，两两图象块的数据进行卷积计算。因为每两个二维坐标可以得到一个关联值，所以最后结果是四维的。



优化步骤使用的是“上卷积层”（Upconvolutional layer），包含一个上池化层（unpooling）和一个卷积层。对特征图使用上卷积后与网络收缩的部分结合，最后用一个上采样的粗糙光流预测（upsampled coarser flow prediction），得到最后的结果。这样既保留了粗糙的特征图中的全局信息也保留了低层特征图中的局部信息。

1.1.2 FlowNet 2.0:



FlowNet2.0 在 FlowNet 的基础上提升了性能,达到了业界领先水平。这是因为 FlowNet2.0 解决了在估计光流场过程中小位移与噪声人造制品（noisy artifacts）的问题，所以在动作识别与运动分割的应用表现上有很大提升。

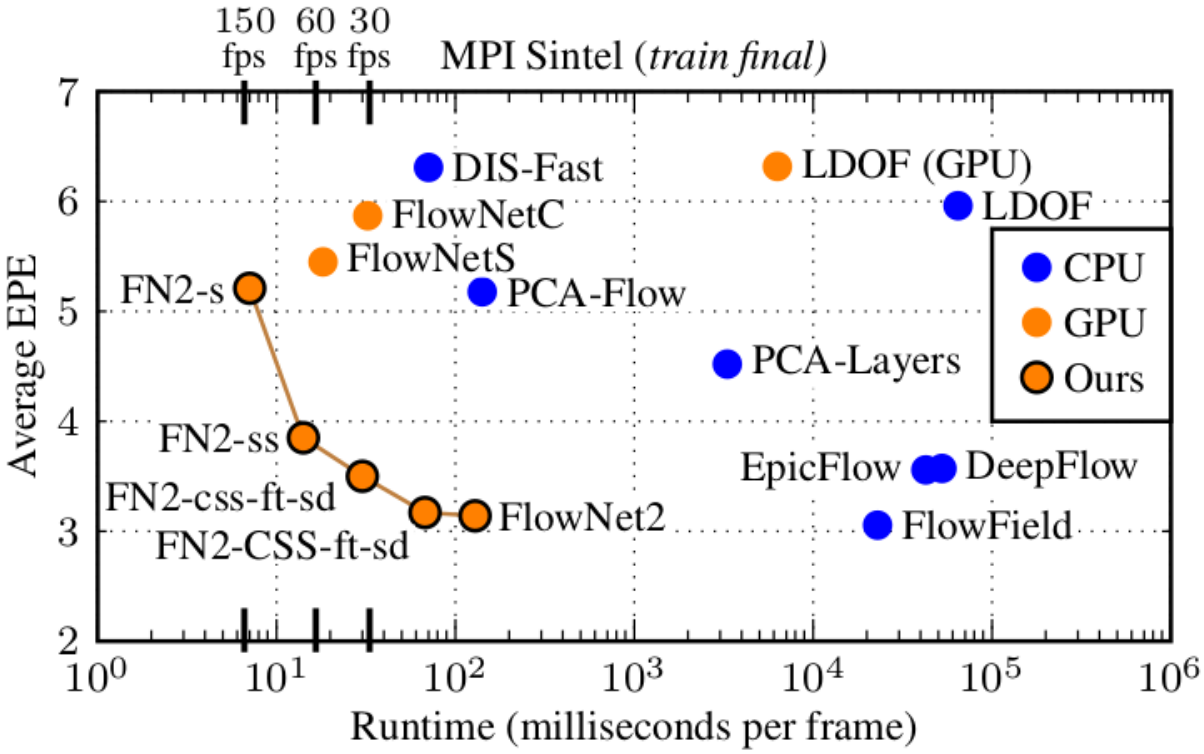
FlowNet2.0 的具体改进有三点：首先是改变了训练数据的顺序。作者发现单独使用复杂数据的效果较差，而多个从简单到复杂的训练数据集能显著地提高效果。而且与 FlowNet 论文中的结论不同的是，作者发现 FlowNetCorr 比 FlowNetSimple 的表现要好得多。

其次是引进了一个扭曲的操作来堆叠多个网络。通过改变堆的深度和每个网络的大小得到了很多不同大小和运行时间的网络。所以 FlowNet2.0 可以控制精度和计算资源之间的交换。网络可以在 8fps 到 140fps 之间的范围运行。

最后通过创建一个特殊的训练数据集和网络来改善在亚像素层级的动作识别。再把第二步的堆叠网络与这个特殊网络结合来得到对任意动作都有最佳的效果。

1.2 Performance:

复现采用了 Pytorch 实现（代码链接）。为了节省训练时间，直接使用了基于 Caffe 实现的网络，再利用 FlyingChairs 和 FlyingThings3D 数据集进行训练得到了预训练模型。在预训练基础上对 MPI Sintel Train Final 数据集利用 GPU Geforce RTX 2080 进行运算。结果为：**Average EPE=3.49, Average L1 Loss=2.21, mean inference time=85.58ms**。下图为原论文在 MPI Sintel Train Final 数据集上运行结果与其他方法的对比：



由于堆叠网络，如图中曲线函数所示 FlowNet2 可以实现精度与速度的交换，原论文的结果为: Average EPE=3.14, mean inference time=123ms。复现结果比原论文结果的精度要低，但速度要快，符合图中的双曲线函数图像。

2 PWC-Net

2.1 Introduction

PWC-Net 是一种端到端的用于单目光流估计的学习框架。其特点是**特征提取金字塔(Feature pyramid)**。PWC-Net 利用光流估计去改变第二幅图的 CNN 结构，之后利用两幅图的特征构建代价立方体进行光流估计。

2.2 Method

第一部分为两个特征提取金字塔，其分别输入一帧图像到金字塔顶层，并逐次通过卷积层下采样提取特征。利用双线性插值进行变化，并计算特征的梯度进行反馈。

第二部分为变换层（Warping layer）,利用 X2 上采样将第二帧图像的特征金字塔底层变换到第一帧图像上。

第三部分为代价立方体层，计算两帧图像特征的匹配代价。其中匹配代价定义为第一帧图像和变换后的第二帧图像特征之间的相关系数（向量的夹角余弦）。由于进行了卷积，仅需计算小范围的像素。

第四部分为光流估计，其为一多层 CNN，输入代价立方体、第一幅图的特征、及上采样光流图，输出金字塔某一层上的权重。各层权重及参数相互独立。估计层若利用全连接层则可以更好地进行图像识别。

第五部分为背景层，利用一个称作背景网络（Context Network）来扩大接受域（Receptive Field）。其结构为一个前馈 dilated CNN，包含 7 个卷积层。

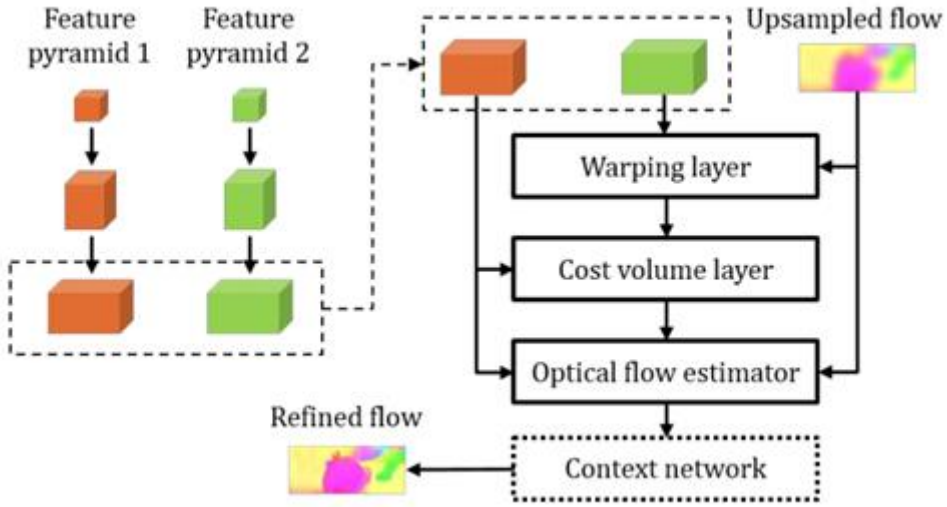


图 1 PWC-Net 结构概览

2.3 Performance

网络训练利用的是 FlyingChairs 数据集[4],并利用 FlyingThings3D 数据集（排除了运动速度过于极端的情况）调整模型，最后利用 Sintel 或 KITTI 调整模型。

SIntel 数据集的结果如下：

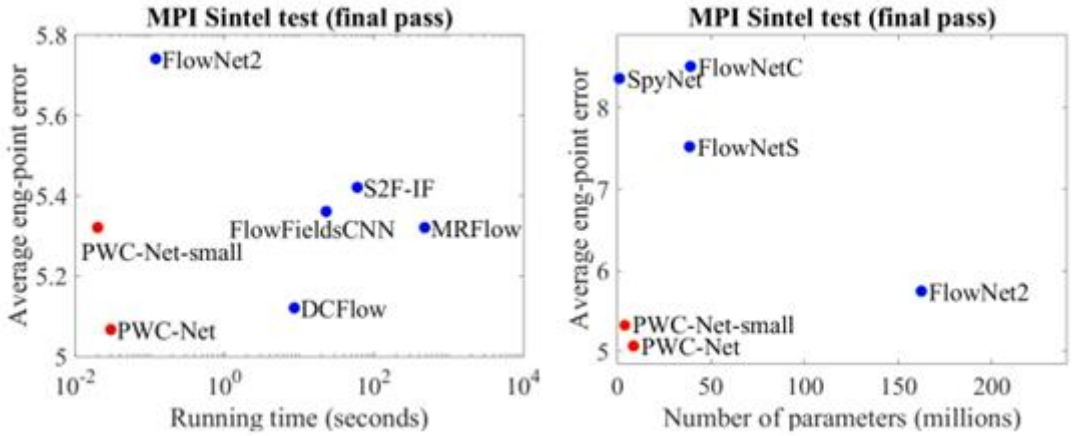


表 1 MPI Sintel test benchmark（Final Pass)

Methods	Training		Test		Time
	Clean	Final	Clean	Final	(s)
PatchBatch	-	-	5.79	6.78	50.0
EpicFlow	-	-	4.12	6.29	15.0
CPM-flow	-	-	3.56	5.96	4.30
FullFlow	-	3.60	2.71	5.90	240
FlowFields	-	-	3.75	5.81	28.0
MRFlow	1.83	3.59	**2.53**	5.38	480
FlowFieldsCNN	-	-	3.78	5.36	23.0
DCFlow	-	-	3.54	5.12	8.60
SpyNet-ft	(3.17)	(4.32)	6.64	8.36	0.16
FlowNet2.0	2.02	3.14	3.96	6.02	0.12
FlowNet2.0-ft	(**1.45**)	(**2.01**)	4.16	5.74	0.12
PWC-Net-small	2.83	4.08	-	-	**0.02**
PWC-Net-small-ft	(2.27)	(2.45)	5.05	5.32	**0.02**
PWC-Net	2.55	3.93	-	-	0.03
PWC-Net-ft	(1.70)	(2.21)	3.86	5.13	0.03
PWC-Net-ft-final	(2.02)	(2.08)	4.39	**5.04**	0.03

表 2 Results on the Sintel benchmark

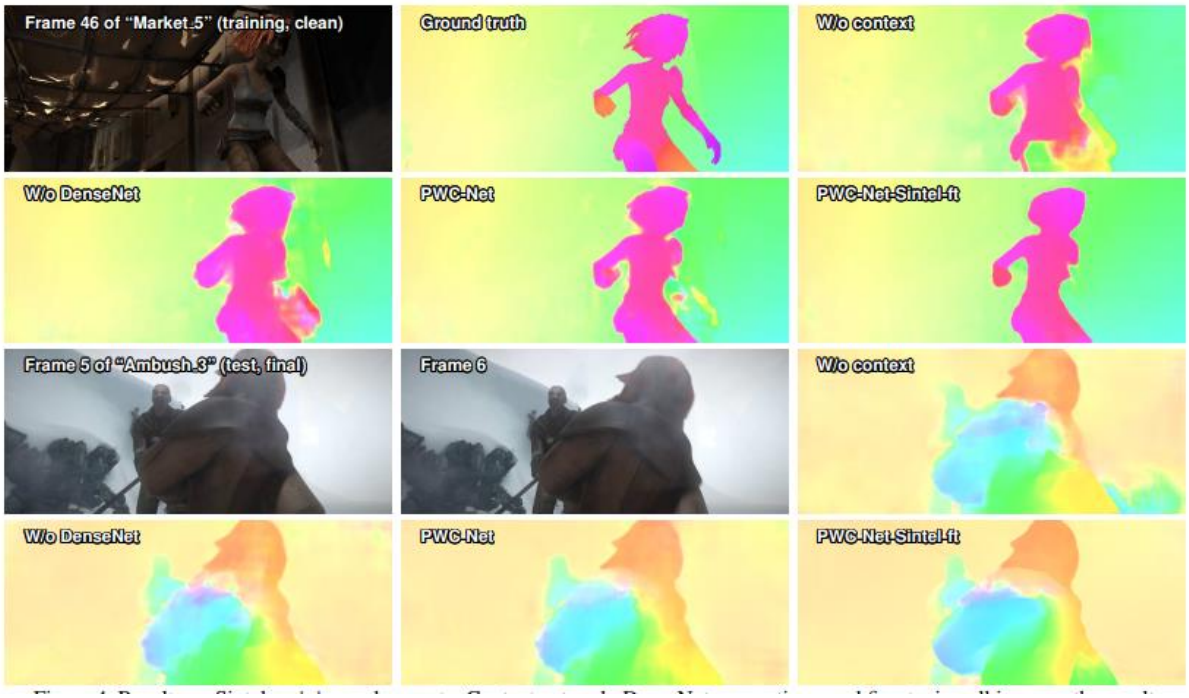


图 2 Result on Sintel sets

由表 1、表 2 可以看到 PWC-Net 在 MPI Sintel Final Pass 上的效果非常好，其运算速度优于文章发表时的所有在 MPI Sintel final pass 数据集性能测试上公开过的所有算法。PWC-Net-small 网络为 PWC-Net 减少其全连接层的连接后得到，其运算速度比 PWC-Net 快 40%，而只减小了 5%的精确度。PWC-Net 在 1024 X 436 的 分辨率上训练速度可达到 35 FPS，而在参数数目方面，PWC-Net 的参数数目远小于其他方法。在图 2 中可以看到 PWC-Net 可以清

楚地描绘运动物体的轮廓，但是对于小的、快速运动的物体无法描绘其轮廓，如 Market-5 中的左手手臂。也可以看到 Context-Net 等在锐化轮廓上的作用十分显著。

进一步的实验结果是 PWC-Net 在 MPI Sintel Clean Pass（不渲染动态模糊、烟雾等）上比传统算法误差更大，这是因为传统方法利用图像边界来进行预测。而在 Final Pass（渲染动态模糊、烟雾等）上，PWC-Net 更加精确，这表明其处理实际图像时性能更好。同时其在训练集上的误差比 FlowNet2 大，但在测试集上反而小，其泛化能力更强。

对于 KITTI 数据集，结果如下：

	KITTI 2012			KITTI 2015		
Methods	AEPE *train*	AEPE *test*	Fl-Noc *test*	AEPE *train*	Fl-all *train*	Fl-all *test*
EpicFlow [39]	-	3.8	7.88%	-	-	26.29 %
FullFlow [14]	-	-	-	-	-	23.37 %
CPM-flow [21]	-	3.2	5.79%	-	-	22.40 %
PatchBatch [17]	-	3.3	5.29%	-	-	21.07%
FlowFields [2]	-	-	-	-	-	19.80%
MRFlow [53]	-	-	-	-	14.09 %	12.19 %
DCFlow [55]	-	-	-	-	15.09 %	14.83 %
SDF [1]	-	2.3	**3.80%**	-	-	11.01 %
MirrorFlow [23]	-	2.6	4.38%	-	9.93%	10.29%
SpyNet-ft [38]	(4.13)	4.7	12.31%	-	-	35.07%
FlowNet2 [24]	4.09	-	-	10.06	30.37%	-
FlowNet2-ft [24]	(**1.28**)	1.8	4.82%	(2.30)	(**8.61%**)	10.41 %
PWC-Net	4.14	-	-	10.35	33.67%	-
PWC-Net-ft	(1.45)	**1.7**	4.22%	(**2.16**)	(9.80%)	**9.60%**

表 3 Result on KITTI sets

图 3 Result on KITTI sets

可以看到其同样具有清晰的边界，快速运动的较大物体轮廓也很清晰。截至论文发布时，PWC-Net 是 KITTI 2015 上全局及非遮蔽区离群流（flow outliers）最少的，在 KITTI 2012 上只有 SDF 在非遮蔽区的性能超过了它，这是因为 SDF 假设了背景为刚体，而 KITTI 2012 大多为静态背景，而 KITTI 2015 大多为动态背景。

模型大小及利用 NVIDIA TitanX 的训练时间如下表：

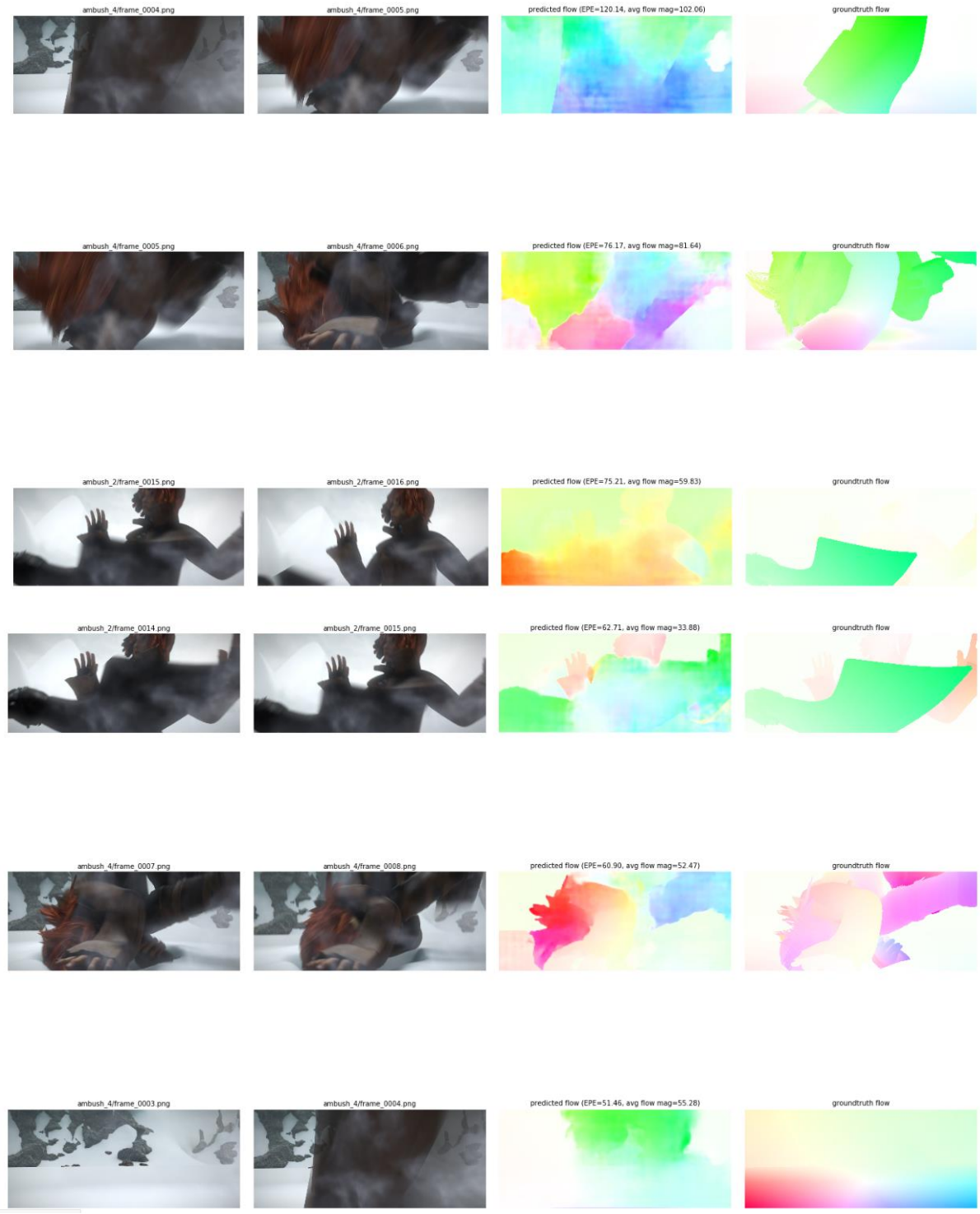
Methods	FlowNetS	FlowNetC	FlowNet2	SpyNet	PWC-Net	PWC-Net-small
#parameters (M)	38.67	39.17	162.49	1.2	8.75	4.08
Parameter Ratio	23.80%	24.11%	100%	0.74%	5.38%	2.51%
Memory (MB)	154.5	156.4	638.5	9.7	41.1	22.9
Memory Ratio	24.20%	24.49%	100%	1.52%	6.44%	3.59%
Training (days)	4	6	>14	-	4.8	4.1
Forward (ms)	11.40	21.69	84.80	-	28.56	20.76
Backward (ms)	16.71	48.67	78.96	-	44.37	28.44

表 4 Model size and running time

可以看到 PWC-Net 训练速度至少是 FlowNet2 的三倍且内存占用只为前者的十五分之一。而实际上 PWC-Net-small 的训练时间只比 PWC-Net 少一些，但是其内存占用少了接近一半。

FlowNet2 通过堆叠基本模型获得了很好的效果，而 PWC-Net 则利用更小的模型达到了同样甚至更好的效果，这是因为 PWC-Net 嵌入了一些经典的原理。

为了复现文中的效果，采用了 TensorFlow 实现，利用 FlyingChairs 和 FlyingThings3D 进行训练得到的预训练模型，直接对 SintelFinal 利用 i7-7700 进行运算，结果为：**Average EPE=3.70, mean inference time=1014.40ms**（根据程序说明书，若利用 GTX 1080 进行运算时间应为 80ms 左右），误差较大的几幅图如下所示：


ambush_4/frame_0004.png
ambush_4/frame_0005.png
predicted flow (EPE=120.14, avg flow mag=102.06)
groundtruth flow
ambush_4/frame_0005.png
ambush_4/frame_0006.png
predicted flow (EPE=76.17, avg flow mag=81.64)
groundtruth flow
ambush_2/frame_0015.png
ambush_2/frame_0016.png
predicted flow (EPE=75.21, avg flow mag=59.83)
groundtruth flow
ambush_2/frame_0014.png
ambush_2/frame_0015.png
predicted flow (EPE=62.71, avg flow mag=33.88)
groundtruth flow
ambush_4/frame_0007.png
ambush_4/frame_0008.png
predicted flow (EPE=60.90, avg flow mag=52.47)
groundtruth flow
ambush_4/frame_0003.png
ambush_4/frame_0004.png
predicted flow (EPE=51.46, avg flow mag=55.28)
groundtruth flow

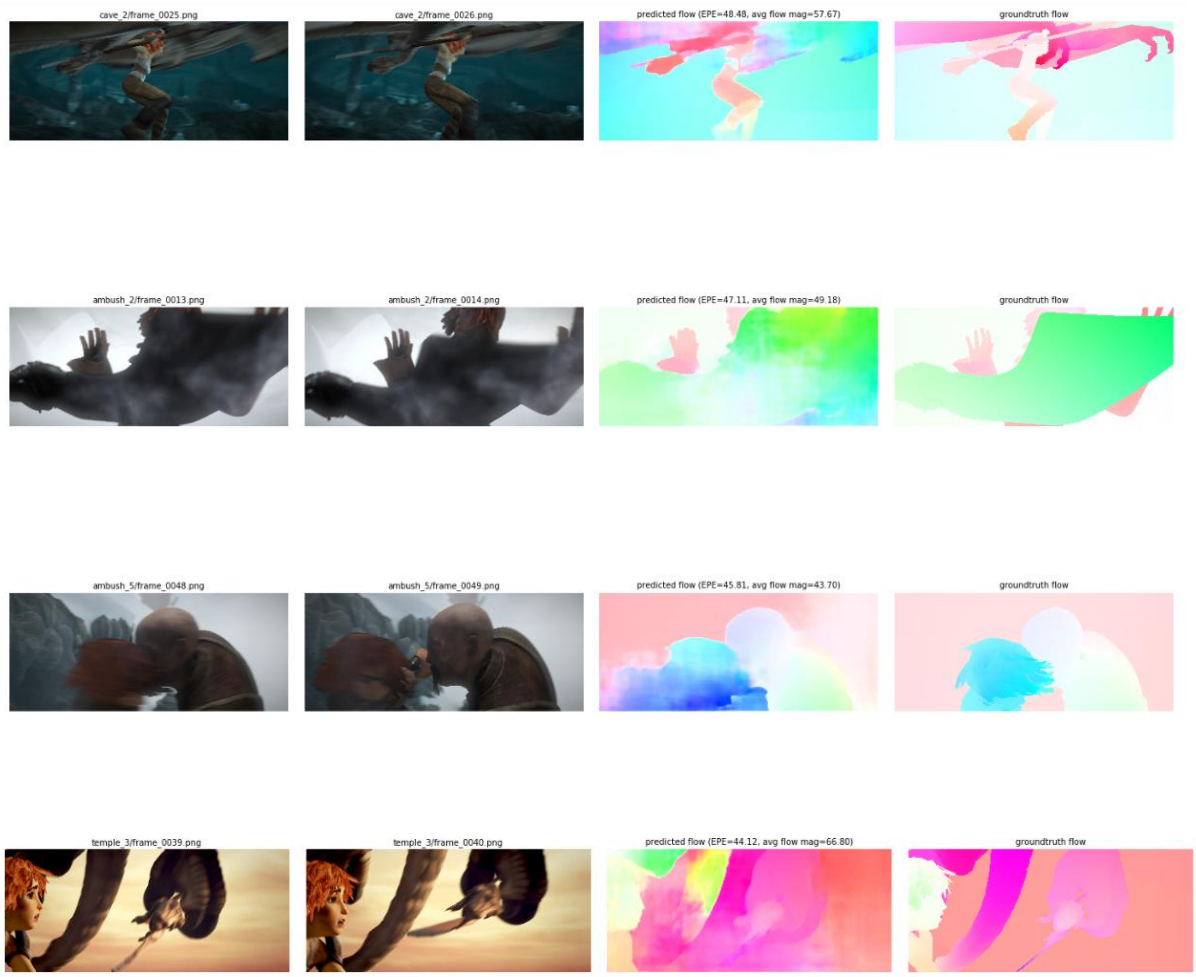


图 4 EPE 最大的 10 对图像

可以看到误差较大的情况有如下几种：①近处高速运动的物体，前后两帧物体变化剧烈，物体部分消失在视野外、部分视野外的物体进入视野内（如第一、第六对图像），导致两帧图像难以互相匹配。②粒子效果的干扰导致误差（如第四、第八对中的雾气）③剧烈动态模糊导致的误差（如第五、第九对图像）

误差最小的 10 对图像如下：


cave_2/frame_0044.png
cave_2/frame_0045.png
predicted flow (EPE=0.64, avg flow mag=16.61)
groundtruth flow
cave_2/frame_0043.png
cave_2/frame_0044.png
predicted flow (EPE=0.73, avg flow mag=23.11)
groundtruth flow
ambush_7/frame_0039.png
ambush_7/frame_0040.png
predicted flow (EPE=0.76, avg flow mag=14.88)
groundtruth flow
market_5/frame_0040.png
market_5/frame_0041.png
predicted flow (EPE=0.88, avg flow mag=23.97)
groundtruth flow
cave_2/frame_0010.png
cave_2/frame_0011.png
predicted flow (EPE=0.92, avg flow mag=27.51)
groundtruth flow

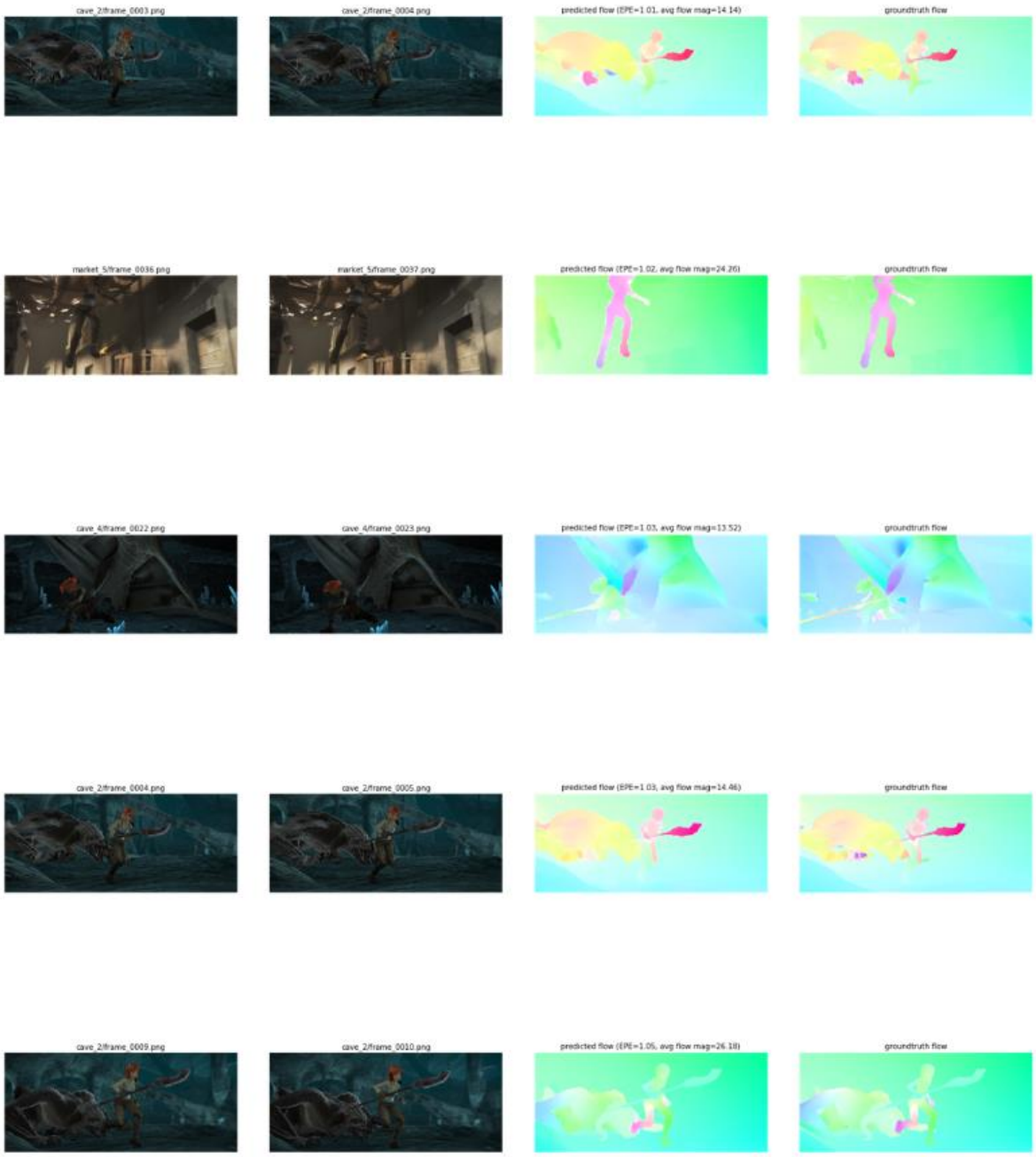


图 5 误差最小的 10 对图像

可以看到 EPE 小的场景都是运动不剧烈、运动物体距离适中、运动物体较少且相互距离较远，场景较为简单，没有太多粒子效果。事实上在实际运用上类似的场景较少，

3 GeoNet

3.1 Introduction

GeoNet 是一种用于视频中单目深度、光流和相机运动估计的无监督学习框架。这三者**通过三维场景几何特性耦合**在一起，以端到端的方式进行**联合学习**。

具体过程为：首先根据静态场景对相机姿态及深度图进行推理，然后用 ResFlowNet 学习剩余的非刚性流(residual non-rigid flow)，从每个单独模块的预测中提取几何关系，合并为图像重构损失。除此之外，本文还提出了一种自适应几何一致性损失损失来提高对离群点和 non-Lambertian 区域的鲁棒性，有效地解决了遮挡和纹理模糊的问题。

在 KITTI 数据集上的实验表明，GeoNet 在三个任务上都分别取得了 state-of-the-art 的结果，比以前的无监督学习的方法表现更好，**甚至可以与监督学习的方相媲美**。

3.2 Method

GeoNet 以无监督学习的方式感知 3D 场景的几何形状。将整个框架分为两个部分——刚性结构重构器（rigid structure reconstructor）&非刚性结构定位器（non-rigid motion localizer），分别来学习刚体流和剩余流（residual flow）。利用图像相似度来引导无监督学习，可以推广到无限数量的视频序列而不需要任何标记成本。

第一部分由两个子网络构成，即 DepthNet 和 PoseNet，分别回归出深度图和相机姿态。由于 DepthNet 推测的是像素级的几何关系，我们采用[5]中的网络结构。这个结构主要由编码器（encoder）和解码器（decoder）两部分组成。编码器部分以 ResNet50 作为更有效的剩余学习方式。解码器由反卷积层组成，将空间特征映射放大到输入的全尺度。为了同时保留全局高层次特征和局部细节信息，在 encoder 和 decoder 之间的不同分辨率上采用了 skip connections，进行了多尺度的深度预测。PoseNet 回归了 6DoF 的相机姿态，欧拉角和平移

向量。 PoseNet 的网络结构与[6]相同，8 个卷积层后连接着一层全局平均池化层，最后是预测层。除了最后的预测层之外，其他层都采用了 Batch 正则化和 ReLUs 激活函数。

一、二部分通过一种级联结构相连，用视图合成损失（view synthesis loss）引导这种融合运动场的无监督学习。

第二部分的 ResFlowNet 用于处理运动的物体,采用和 DepthNet 相同的网络结构。将 ResFlowNet 学习的剩余非刚性流动与刚性流动相结合，得到最终的流动预测流。刚性结构器在第一阶段产生了高质量的重构，为第二阶段奠定了良好的基础，因此 ResFlowNet 只需要关心剩余的非刚体。不仅可以纠正运动物体预测的错误，还可以纠正第一阶段不完美的结果。

最后是自适应几何一致性损失的检测，用来克服那些不包括在 view synthesis 中的目标，比如遮挡和光度不一致的问题。通过模仿传统的前后向(或左-右)一致性检查，过滤了可能的离群点和遮蔽区域。

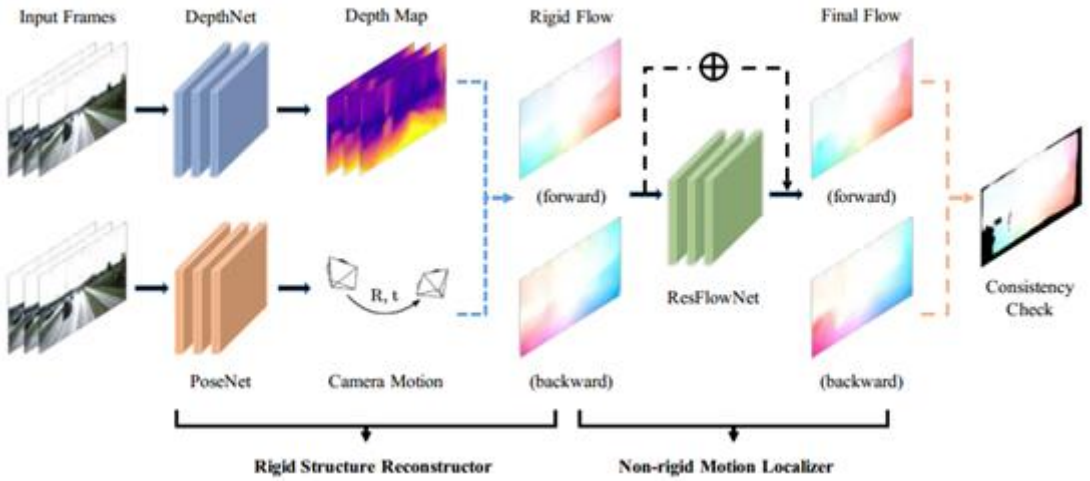


图 1 GeoNet 结构概览

3.3 Performance（Optical Flow）

主要对比 GeoNet 在 KITTI stereo/flow split 上测试光流部分的表现，由于采用无监督学习方式，可以用不带 grountruth 的原始图像进行训练。所有的图像一共包含了 28 个场景（测试图像除外），对于图像分辨率采用线性插值的方式将其变换为 128 X 416。对于运行时间，网

络在 TitanXP 上用于推断每一个测试样例的深度、光流及相机姿态分别用时 14ms、45ms、以及 4ms，但我认为不可单纯地将光流运算时间看作 45ms，因为其利用了深度图和相机位姿的运算结果。对于计算资源文献中尚未提及，。

Method	Dataset	Noc	All
EpicFlow [38]	-	4.45	9.57
FlowNetS [8]	C+S	8.12	14.19
FlowNet2 [18]	C+T	4.93	10.06
DSTFlow [37]	K	6.96	16.79
Our DirFlowNetS (no GC)	K	6.80	12.86
Our DirFlowNetS	K	**6.77**	12.21
Our Naive GeoNet	K	8.57	17.18
Our GeoNet	K	8.05	**10.81**

表 1 Average end-point error over non-occluded regions(Noc) and overall regions(All)

由表可知，GeoNet 在非遮蔽区虽不及其他监督学习的方法，但优于其他非监督学习方法。同时其全局误差可以媲美监督学习方法，因此在 Groundtruth 难以获得的情况下，GeoNet 有其一定的优越性。

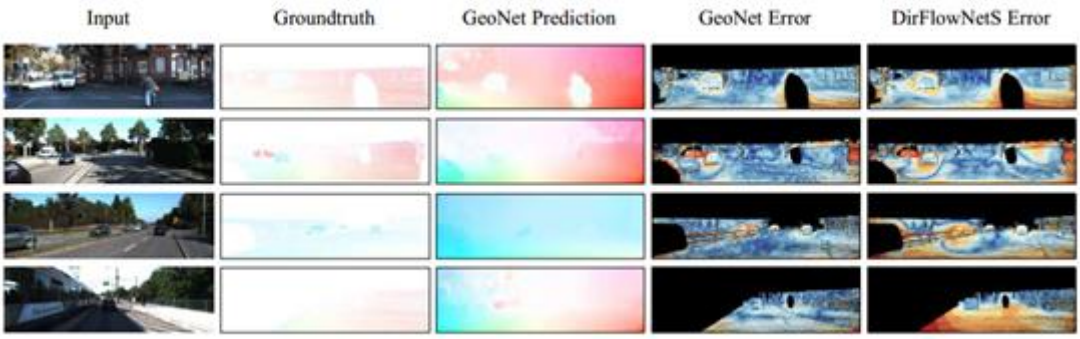


图 2 Comparison of direct flow learning method

由上图可以看到，GeoNet 在遮蔽区、纹理模糊区的误差明显小于 DirFlowNetS。但从表 1 可以看出，其在非遮蔽区性能不如 DirFlowNetS，原因分析如下：

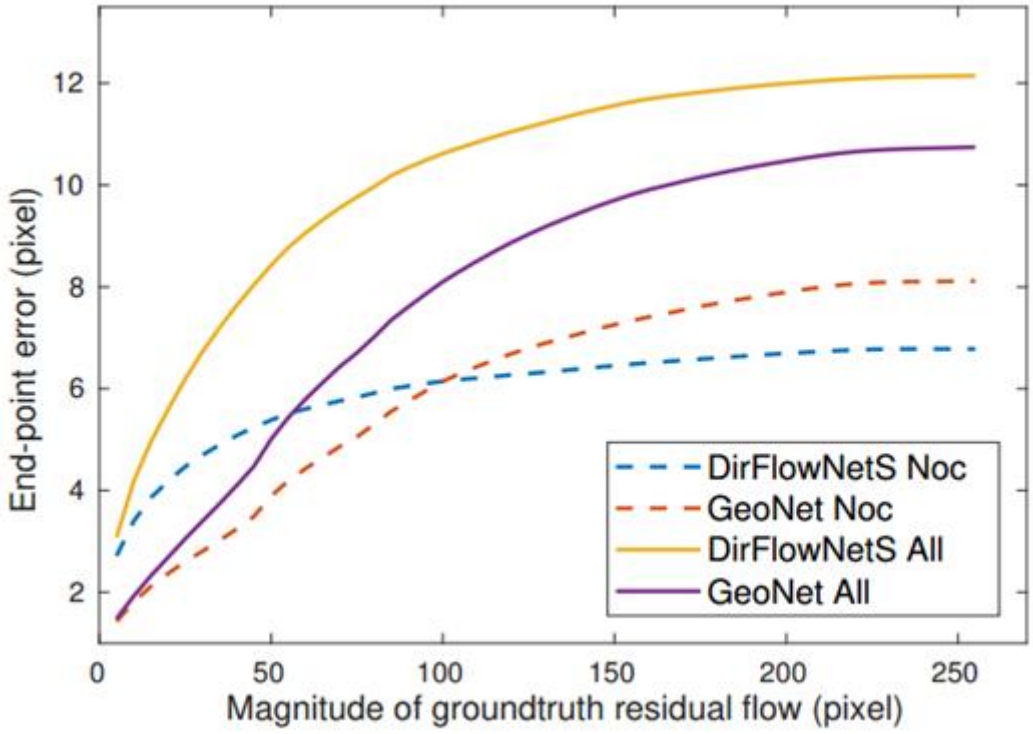


图 3 Average EPE at different magnitude of groundtruth residual flow

通过上图可以看见，随着 Groundtruth 残余流的增加，GeoNet 在非遮蔽区域的 EPE 增加较快，这是因为损失函数是基于像素灰度差值，而级联结构越复杂，其差值越大。但若只将 warping loss 替换为数值监督的损失[7]，则该现象消失。

4 Comparison

4.1 Method Comparison

FlowNet2、PWC-Net、GeoNet 三者都是端到端的方法，其中 GeoNet 是无监督的方法，而后两者为监督学习。同时 GeoNet 还能得到深度图和相机姿态，因其利用了这两者与光流的物理关系进行计算。

对于图像序列的处理上面，GeoNet 分别采用了积分图卷积[5]及 Hessian 矩阵检测器[6]提取深度图、中一种无监督的方法计算相机位姿，再利用两者结果指导光流图的学习，同时其一个图像序列里面包含三帧图像；PWC-Net 利用了两个金字塔型特征提取器对两幅图分别多次进行上采样，之后将第二帧的特征进行变换（warping）后与第一幅图叠加;FlowNet2 采用了 FlowNetS 简单将两图叠加，以及 FlowNetC，先单独提取两幅图特征，再关联匹配两幅图。

在光流图的计算方面，GeoNet 利用深度图及相机位姿分离背景与运动物体，利用视图合成损失指导网络的学习，网络结构与深度图的相同。PWC-Net 通过代价立方体计算光流图，其中匹配代价定义为得到的特征向量的夹角余弦，之后将代价立方体输入到 CNN 里面，金字塔不同层单独训练，最后得到光流图。FlowNet2 一方面将多个编码-解码的网络结构堆叠，另一方面，还专门建立了一个用于辨别小位移物体的网络，最后两者将各自的结果结合起来。

4.2 Performance Comparison

根据 KITTI 及 MPI-Sintel 官方排行，三种方法及其衍生型的性能如下（由于 LiteFlowNet2[8] 和 FlowNet2 之间差距较大，而 PWC-Net+是仅仅改变了训练方式的 PWC-Net，因此主要对比 FlowNet2、PWC-Net+[9]及 GeoNet）：

Rank	Method	Out-Noc	Out-All	Avg-Noc	Avg-All	Runtime	Environment
6	LiteFlowNet2-MD+	2.59%	6.10%	0.7 px	1.4 px	0.058 s	NVIDIA GTX 1080
7	LiteFlowNet2	2.72%	6.30%	0.7 px	1.4 px	0.0486 s	NVIDIA GTX 1080
15	LiteFlowNet	3.27%	7.27%	0.8 px	1.6 px	0.0885 s	NVIDIA GTX 1080
17	PWC-Net+	3.36%	6.72%	0.8 px	1.4 px	0.03 s	NVIDIA Pascal Titan X
31	PWC-Net	4.22%	8.10%	0.9 px	1.7 px	0.03 s	NVIDIA Pascal Titan X
36	GeoFlow	4.40%	12.16%	1.0 px	2.9 px	0.04 s	GPU @ 2.5 Ghz (Python)
43	FlowNet2	4.82%	8.80%	1.0 px	1.8 px	0.1 s	GPU @ 2.5 Ghz (C/C++)

表 1 KITTI2012 上各方法及衍生型性能

- **Out-Noc:** Percentage of erroneous pixels in non-occluded areas
- **Out-All:** Percentage of erroneous pixels in total
- **Avg-Noc:** Average disparity / end-point error in non-occluded areas
- **Avg-All:** Average disparity / end-point error in total
- **Density:** Percentage of pixels for which ground truth has been provided by the method

Rank	Method	Fl-bg	Fl-fg	Fl-all	Runtime	Environment
12	LiteFlowNet2-MD+	0.0736	0.0779	0.0743	0.058 s	NVIDIA GTX 1080
17	PWC-Net+	0.0769	0.0788	0.0772	0.03 s	NVIDIA Pascal Titan X
18	LiteFlowNet2	0.0785	0.072	0.0774	0.0486 s	GTX 1080
27	LiteFlowNet	0.0966	0.0799	0.0938	0.0885 s	GTX 1080
28	PWC-Net	0.0966	0.0931	0.096	0.03 s	NVIDIA Pascal Titan X
32	FlowNet2	0.1075	0.0875	0.1041	0.1 s	GPU @ 2.5 Ghz (C/C++)

表 2 KITTI2015 上各方法及衍生型性能（GeoNet 未提交结果）

- **Fl:** Percentage of optical flow outliers
- **bg**: Percentage of outliers averaged only over background regions
- fg: Percentage of outliers averaged only over foreground regions
- all: Percentage of outliers averaged over all ground truth pixels

相较于 KITTI2012，KITTI2015 有更多的动态场景、遮挡、截断等，因此更具挑战性。考虑到 TitanXPascal 在不超频的情况下性能约为 GTX1080 的 120%[10]，PWC-Net 和 GeoNet 在速度上确实较有优势，但也可以看到 FlowNet2 的衍生型 LiteFlowNet2 的准确度和速度相比 FlowNet2 有相当的提升，尤其是其背景离群点比 FlowNet2 减少近 30%。GeoNet 在计算速度上介于 LiteFlowNet 和 PWC-Net 之间，精度优于 FlowNet2 但不如 PWC-Net，考虑到其是非监督的方法，在 KITTI 数据集上其性能还是可观的。

Rank	Method	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+
7	PWC-Net+	4.596	2.254	23.696	4.781	2.045	1.234	0.945	2.978	26.62
10	LiteFlowNet2-MD+	4.728	2.249	24.939	4.01	1.925	1.504	0.783	2.634	29.369
16	LiteFlowNet2	4.903	2.346	25.769	4.142	2.014	1.546	0.797	2.529	31.039
22	PWC-Net	5.042	2.445	26.221	4.636	2.087	1.475	0.799	2.986	31.07
53	FlowNet2-ft-sintel	5.739	2.752	30.108	4.818	2.557	1.735	0.959	3.228	35.538
61	FlowNet2	6.016	2.977	30.807	5.139	2.786	2.102	1.243	4.027	34.505
138	GeoFlow	8.459	3.908	45.553	5.805	3.550	2.899	1.398	4.667	52.653

表 3 Sintel-Final 上各方法及衍生型性能

- EPE:Endpoint error over the complete frames
- EPE matched:Endpoint error over regions that remain visible in adjacent frames
- EPE unmatched:Endpoint error over regions that are visible only in one of two adjacent frames
- d0-10:Endpoint error over regions closer than 10 pixels to the nearest occlusion boundary

- d10-60:Endpoint error over regions between 10 and 60 pixels apart from the nearest occlusion boundary
- d60-140:Endpoint error over regions between 60 and 140 pixels apart from the nearest occlusion boundary
- s0-10:Endpoint error over regions with velocities lower than 10 pixels per frame
- s10-40:Endpoint error over regions with velocities between 10 and 40 pixels per frame
- s40+:Endpoint error over regions with velocities larger than 40 pixels per frame

Sintel-Final 是带粒子效果、烟雾、动态模糊等特效的人工合成 CG，相较于 KITTI，其物体运动速度更快、方向更多变，也有更多的近处的高速运动物体，但缺少类似汽车外壳的反光表面等。在 Sintel 官方性能测试上，PWC-Net+在各方面都优于 Flow-Net2 和 GeoNet，而 GeoNet 相较于其在 KITTI 数据集上，其性能大大降低，其原因为其作为无监督的方式，对物体的截断、动态模糊、烟雾等特殊情况鲁棒性较差。实际利用预训练模型进行复现时

Methods	FlowNetS	FlowNetC	FlowNet2	SpyNet	PWC-Net	PWC-Net-small
#parameters (M)	38.67	39.17	162.49	1.2	8.75	4.08
Parameter Ratio	23.80%	24.11%	100%	0.74%	5.38%	2.51%
Memory (MB)	154.5	156.4	638.5	9.7	41.1	22.9
Memory Ratio	24.20%	24.49%	100%	1.52%	6.44%	3.59%
Training (days)	4	6	>14	-	4.8	4.1
Forward (ms)	11.40	21.69	84.80	-	28.56	20.76
Backward (ms)	16.71	48.67	78.96	-	44.37	28.44

表 4 FlowNet 及 PWC-Net 计算资源

从上表中可以看到 FlowNet2 比较臃肿，需要的内存占用大、训练时间长，若对这两方面比较敏感（如移动端）则应选择 PWC-Net。

4.3 Result

FlowNet2 给了我们启示，将网络以某种形式堆叠可以降低光流图误差，而且其堆叠具有灵活性，可以根据需要增减网络，以在速度和精度之间达到一个满足需求的平衡点，同时科学合理安排训练集、训练顺序也能提高精度，正如提出 PWC-Net+的文献中所阐述的，将不同的模型直接相比一定程度上是不公平的，因为某些模型的潜力可能因为训练不当而未被完全发掘。而这篇文献也只是一种经验上的研究（Empirical Study），但其提出的 PWC-Net+确实在准确

度上提升了 10%~20%。GeoNet 则是利用几何空间及运动速度上的关系将深度图、相机姿态、光流图三者结合在了一起，用前两者的结果来指导光流图运算，而不仅仅是利用大量训练数据在数值上去逼近模型。同时也可以看到，在不同数据集上各方法性能差异较大，KITTI 虽然是在真实环境下的测量结果，但其都是在车辆较少、车速较慢的情况下进行测量，且没有特殊情况下的结果（如小孩突然从路边跑出），而这些是最考验算法的安全性的情况。而 Sintel 虽有近处高速运动物体，但其是合成的动画，与真实环境还是有所差异。

总的来说，PWC-Net+速度快、结构精简。FlowNet2 速度较慢、结构较为臃肿。作为非监督的方法，GeoNet 在 KITTI2012 这样较为简单的场景下精度、速度都较好，但复杂环境下其准度较差。实际上虽然 FlowNet 开创性地将 CNN 成功运用到光流估计中，但其结构臃肿，若要考虑实际运用，我认为还是应该考虑更为精简的 PWC-Net 及 FlowNet2 的衍生型 LiteFlowNet2。

Reference

[1]E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[2]Sun, Deqing , et al. "PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume." (2017).

[3]Yin, Zhichao , and J. Shi . "GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose." (2018).

[4].Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In ACM Multimedia,2014.

[5]H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *CVIU*, 2008.

[6]T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, 2017.

[7]G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. CoRR, 2015.

[8]LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation

[9]Models Matter, So Does Training: An Empirical Study of CNNs for Optical Flow Estimation

[10]https://gpu.userbenchmark.com/Compare/Nvidia-Titan-X-Pascal-vs-Nvidia-GTX-1080/m158352vs3603