浙江工业大学

本科毕业设计论文

外文翻译

题目：双目视觉立体匹配算法设计与实现

作者姓名 王 灏

指导教师 宣琦研究员

专业班级 通信工程 1301

学　　院 信息工程学院

提交日期 2017 年 2 月 28 日

一个训练视差，光流，场景流估计卷积网络的大型数据集

Philip Hausser ，Philipp Fischer ，Daniel Cremers，慕尼黑工业大学

Nikolaus Mayer，Eddy Ilg，Alexey Dosovitskiy, Thomas Brox，弗赖堡大学

摘要:最近的工作表明，光流估计可以被制定为监督学习的任务，并且可以用卷积网络解决。训练所谓的 FlowNet 是由大型的综合生成的数据集实现的。本文将利用卷积网络进行光流估计的概念扩展到视差和场景流估计中。为此，我们提出三个合成立体视频数据集，它们具有充足的真实性，变化情况和大小范围来成功地训练大型网络。我们的数据集是第一个允许训练和评估场景流方法的大规模数据集。除了数据集，我们还提出了一个实时视差估计的卷积网络来提供最先进的结果。通过组合流和视差估计网络并对其进行训练它们，我们演示了第一个使用卷积网络的场景流估计。

1．介绍

估计场景流意味着提供立体视频中所有可见点的深度和 3D 运动矢量。当谈到重建和运动估计时，它是“皇家联盟”任务并提供了一个重要的基础给很多更高层次的挑战，例如先进的驾驶员的辅助系统和自主系统。最近几十年来的研究一直专注于它的子任务，即视差估计和光流估计，并取得了相当大的成功。整个的场景流问题还没有开发到这种程度。既然局部场景流可以简单的从子任务结果中集合出来，我们可以猜测所有组件的联合估计对效率和精确率都是有利的。场景流比其子任务更少被探究的原因之一似乎是缺乏完全注释的地面实况数据。

在卷积网络的时代，这种数据的可用性已经变得更加重要。论文 Dosovitskiy [4]表明光流估计可以看作一个监督学习问题并且可以用一个大型网络解决。为了训练他们的网络，他们创造了一个简单的合成的飞行椅的 2D 数据集并被证明足以预测一般视频的精确的光流。这些结果表明，视差和现场流可以由卷积网络理

想地联合的有效的实时的估计出来。实现这个想法缺少的是一个具有足够真实性和变化情况的大型数据集来训练这样的网络并评估它的性能。

图 1: 我们的数据集提供超过 35000 个立体帧，带有光流，视差和视差变化以及诸如对象间隔的其他数据的密集的标签数据

在本文中，我们提出了三个这样数据集的集合，它们是通过使用开源的 3D 创建套件 Blender 的自定义版本创建的。我们的努力在本质上是和 Sintel benchmark[2]类似的。与 Sintel 的工作相比，我们的数据集足够大来帮助训练卷积网络，而且它为场景流提供了地面实况。特别地，它包括立体彩色图像和双向视差，双向光流，视差变化，运动边界和对象间隔的地面真实情况。此外，相机完全标定和 3D 点定位是可用的，即我们的数据集也包括 RGBD 数据。

我们不能在单篇论文中利用这个数据集的全部潜力，但我们已经展示了各种与卷积网络训练相结合的用途示例。我们训练一个用于视差估计的网络，其在以前的基准上产生了有竞争力的性能，特别是在那些实时运行的方法中。最后，我们还提出了一个场景流估计的网络并提供在一个足够大的测试集上全场景流的第一个定量数字。

表 1: 可用数据集的比较：我们的新集合提供了比任何现有选择更多的注释数据和更多的数据变化。我们的所有数据具有完全连续，密集，准确的地面实况。

Dataset	MPI Sintel [2]	KITTI Benchmark Suite [17]		SUN3D[27]	NYU2[18]	Ours		
		2012	2015			FlyingThings3D	Monkaa	Driving
#Training frames	1 064	194	800	2.5M	1 449	21 818	8 591	4 392
#Test frames	564	195	800	—	—	4 248	—	—
#Training scenes	25	194	200	415	464	2 247	8	1
Resolution	1024×436	1226×370	1242×375	640×480	640×480	960×540	960×540	960×540
Disparity/Depth	✓	sparse	sparse	✓	✓	✓	✓	✓
Disparity change	✗	✗	✗	✗	✗	✓	✓	✓
Optical flow	✓	(sparse)	(sparse)	✗	✗	✓	✓	✓
Segmentation	✓	✗	✗	(✓)	✓	✓	✓	✓
Motion boundaries	✓	✗	✗	✗	✗	✓	✓	✓
Naturalism	(✓)	✓	✓	✓	✓	✗	✗	(✓)

2. 相关工作

数据集。第一个做出重要的努力创建标准数据集的是用于立体视差估计[22]和光流估计[1]的 Middlebury 数据集。Middlebury 立体数据集由真实场景组成的，光流数据集则是真实场景和渲染场景的混合。两个数据集在今天看来都非常小。特别是小测试集会导致严重的手动过拟合。立体数据集的一个优点是可用于相关真实的场景，特别是在自 2014 年以来的最新的高分辨率版本[21]。

MPI Sintel [2]是一个完全合成的数据集来源于一个短的开源动画 3D 电影。它提供了光流的密集地面实况。最近,一个视差的测试版本可用于训练。Sintel 数据集有 1064 个训练帧是目前最大的可用数据集。它包含足够的现实场景，其中包括自然的图像退化例如雾和运动模糊。作者将大量的努力投入到地面实况的所有帧和像素的矫正中。这使得这个数据集成为一个非常可靠的测试集用来进行算法的比较。然而，对于卷积网络的训练，数据集仍然太小。

KITTI 数据集制作于 2012 年[8]并在 2015[17]得以扩展。它包含来自安装在汽车上的已校准的一对相机的道路场景的立体视频。光流和视差的地面实况信息通过 3D 激光扫描仪结合汽车的运动数据获得。虽然这个数据集包含真实数据，但是采集方法将地面实况信息限制在场景的某个静态部分。此外，激光器仅提供高达某一特定距离和高度的稀疏数据。至于最新的版本，汽车的 3D 模型与点云拟合以获得更致密的标签并且包括了移动物体。但是，这些区域的地面实况仍然是一个近似值。

Dosovitskiy 等人[4]为光流估计训练的卷积网络，数据集是在移动的 2D 椅子图像叠加自然背景图像的合成数据集。这个数据集很大，但仅限于单视图光流。

它不包含 3D 动作，并且尚未公开可用。

最新的 Sintel 数据集和 KITTI 数据集都可以用于估计具有一些限制的场景流。在遮挡区域（在一帧中可见，但在另一帧中不可见），场景流的地面实况不可用。在 KITTI 中，场景流最有趣的部分是 3D 点的运动，这部分是缺失的或者是通过拟合汽车 CAD 模型近似的。一个关于最重要的可比数据集及其特征的全面概述在表 1 中给出了。

卷积网络。卷积网络[16]已经证明对于各种识别任务是非常成功的，如图像分类[15]。最近卷积网络的应用还包括单个图像的深度估计[6]，立体匹配[28]和光流估计[4]。

Dosovitskiy 等人的 FlowNet [4]是和我们的工作最相关的。它使用一个编码器--解码器架构，带有额外的收缩和扩展网络部分的交叉链路，其中编码器计算来自增大范围的接收场的抽象特征，解码器通过扩展的上卷积架构[5]重建原始分辨率。 我们采用这种方法来视差估计。

Zbontar[28]等人的视差估计方法使用 Siamese 网络来计算图像块之间的匹配距离。为了实际估计视差，作者执行了基于交叉的代价聚合[28]和半全局匹配（SGM）[11]。与我们的工作相比，Zbontar 等人没有端到端的对视差估计任务训练卷积网络，相应的后果是计算效率和优雅。

场景流。虽然有数百篇关于视差和光流估计的论文，只有几篇是关于场景流的。它们之中没有人使用深度学习的方法。

场景流估计第一次通过 Vedula 等人的工作[23]被推广，他们分析了不同的可能的问题设置。后来的工作是由各种变分方法主导的。Huguet 和 Devernay [12]制定利用联合变分法的场景流估计。Wedel 等人[26]遵循变分框架但是分离视差估计已获得更高的效率和准确性。Vogel 等人[25]利用使用分段刚性模型正则化的超像素分割组合场景流估计的任务。Quiroga 等人[19]扩展正则化矩阵到刚性运动的平滑场。像 Wedel 等人[26]那样，他们分离视差估计并通过 RGBD 视频的深度值来代替。

在 KITTI 的场景流前七名中最快的方法是 Cech 等人[23]，运行时间为 2.4 秒。他们方法采用种子生长算法同时视差和光流估计。

3．场景流的定义

光流是真实世界中 3D 运动到图像平面上的投影。通常情况下，场景流被认为是从立体视频或 RGBD 视频计算潜在的 3D 运动场。假设立体对的两个连续的时间帧 t 和 t + 1，产生四个图像（I_L^t， I_R^t，I_L^{t+1}，I_R^{t+1}）。场景流提供给在四个图像之一的每个可见点的 3D 点定位及其 3D 运动矢量[24]。

这些 3D 量只能在已知的相机内在和外在情况下计算。一个相机关于场景流的独立定义是通过分离光流，视差和视差变化分量[12]获得的，见图 2。这种表述在这个意义上得以完善——如果相机参数是已知的，可见的 3D 点和他们的 3D 运动向量可以从上述分量中计算出来。

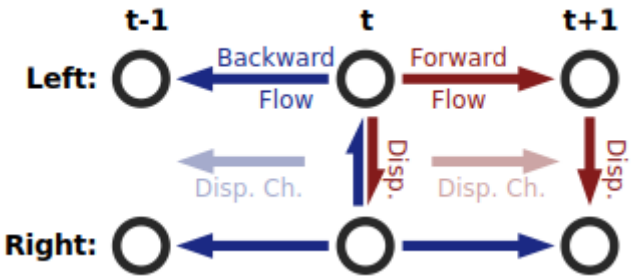

图 2: 给定时间 t-1，t 和 t + 1 的立体图像，箭头表示它们之间的视差和光流关系。红色的箭头通常用于估计场景流。在我们的数据集中，我们提供所有的关系，包括蓝色箭头。

给定 t 和 t + 1 的视差，视差变化几乎可以忽略。因此，在 KITTI 2015 场景流基准中[17]，只评估光流和视差。在这种情况下，场景流只能对同时在左侧和右侧帧可见的表面点重建。特别是在卷积网络的背景下，在部分遮挡区域估计深度和运动是特别有趣的。此外，从流和视差中重建 3D 运动是对噪声更敏感的，因为光流中的一个小误差可能导致 3D 运动矢量中的巨大误差。

4．三个已渲染的数据集

我们创建了一个由三个子集组成的合成数据集套件并提供了完整的前后方向的地面实况场景流（包括视差变化）。为此，我们使用了开源 3D 创建套件 Blender 来动画绘制大量复杂运动的对象和使结果呈现出成千上万帧。我们修改了 Blender 的内置渲染引擎的流水线生成——除立体 RGB 图像外——逐帧逐视

图的三个额外的数据传递。这些数据提供了所有可见表面点的 3D 位置，以及它们未来和过去的 3D 位置。两个这样的为给定的相机视图结果的数据传递之间的像素差异——在 3D 运动矢量的“图像”中——完整的场景流地面实况正如这台相机所见。 注意， 即使在遮挡区域中信息也是完整的，因为渲染引擎总是知道所有的（可见的和不可见的）场景点。所有的透光材料——明显地，大多数车窗——是被渲染为完全透明的以避免 3D 数据中的一致性问题。

给定相机的内在参数（焦距，主点）和渲染设置（图像大小，虚拟传感器尺寸和格式），我们投影每个像素点的 3D 运动矢量到与成像平面共面的 2D 像素运动向量：光流。深度是直接从像素的 3D 位置检索的，并通过虚拟立体视觉平台的已知配置转换为视差。我们从 3D 运动矢量的深度分量中计算出视差变化。结果的示例如图 1 所示。

此外，我们渲染了对象的分割掩码，其中每个像素值对应唯一的对象索引。对象可以由多个子分量组成，其中每个都可以有一个单独的材料（有自己的外在性质例如纹理）。我们利用这一点并渲染额外的分割掩码，其中每个像素编码其材料的索引。最近可用的 Sintel 的测试版本也包括这个数据。

类似于 Sintel 数据集，我们还提供运动边界，在两个或两个以上的移动对象间加亮像素，如果以下条件成立：两个帧之间的运动差异至少为 1.5 个像素， 边界段覆盖至少 10 个像素的区域。阈值是根据 Sintel 的分割结果选定的。

对于所有的帧和视图，我们提供相机完整的内在和外在模型。那些模型可以用于运动的或者其他任务的需要相机追踪的结构。我们使用虚拟的 35mm 焦距的 32mm 宽的模拟传感器来渲染所有的图像数据。为了 Driving 数据集，我们添加了一个焦距为 15mm 的广角版本，其在视觉上更接近现有的 KITTI 数据集。

像 Sintel 数据集，我们的数据集还包括两个每个图像的特别版本：clean pass 版本显示颜色，纹理和场景光照但没有图像退化，而 final pass 版本还包括后处理工作诸如模拟的景深模糊，运动模糊，阳光眩光和伽马曲线操作。

为了处理大量的数据（2.5 TB），我们将所有 RGB 图像数据压缩到有损但高质量的 WebP 格式。非 RGB 数据通过 LZO 无损压缩。

4.1 FlyingThings3D

新数据集的主要部分由日常物体沿着随机的 3D 轨迹飞行组成。我们生成了约 25 000 个带地面实况数据的立体帧。我们不是专注于一个特定的任务（如 KITTI）或执行严格的自然主义（如 Sintel），而是依赖于随机性和一个巨大的渲染资源池来生成数量比任何现有的选项更多的数据，而且不存在重复或饱和的风险。数据生成是快速的，全自动的，并为完整的场景流任务产生密集精确的地面实况。创造这个数据集的动机是为了方便训练得益于大量的种类的大卷积网络。

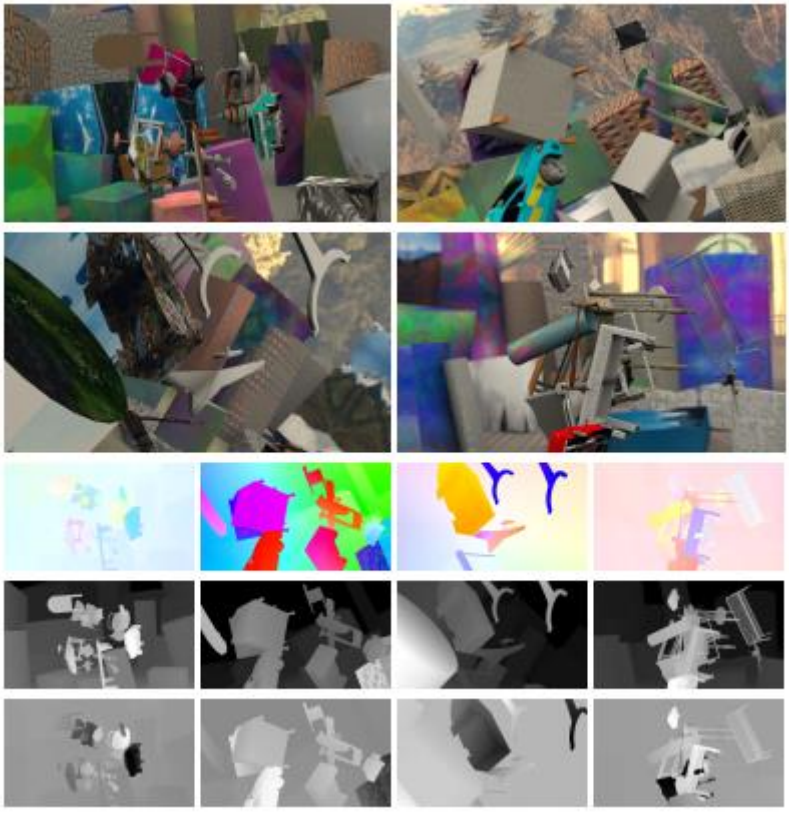

图 3: 来自我们的 FlyingThings3D 数据集的示例场景。**第三行：**光流图，**第四行：**视差图，**第五行：**视差变化图。最好在高分辨率的彩色屏幕上观看（图像归一化显示）。

每个场景的基础是一个大的纹理地平面。我们生成了 200 个静态背景对象 ，它们的形状是从长方体和圆柱体中随机选择。每个对象随机缩放，旋转，纹理化然后放置在地平面上。

要填充场景，我们下载了 35927 个来自斯坦福 ShapeNet 数据库的详细的 3D 模型[20]。从这些模型中，我们组合了一套 32872 个训练模型和一组大小为 3055 的测试模型。当然，模型的类别是不相交。

我们从来自这个对象集的 5 到 20 个的随机对象中进行采样，并随机的纹理化每个对象的每个材料。每个 ShapeNet 对象都已转化并沿着平滑的 3D 轨迹旋转建模，使得相机可以看到对象，但带有随机的位移。相机也是移动的。

纹理集的组成通过 ImageMagick 创建的过程图像，来自 Flickr 的自然风景和城市景观照片以及来自 Image * After 的纹理样式图像。像 3D 模型一样，纹理被分为不相交的训练和测试部分。

对于最终的通过图像，场景存在不同强度的运动模糊和散焦模糊。

4.2 Monkaa

我们的数据集的第二部分来自开源的 Blender 的动画短片 Monkaa 资源。在这方面，它类似于 MPI Sintel 数据集。Monkaa 包含非刚性和软关节运动以及可见的有挑战性的皮毛。除此之外，几乎没有可见的与 Sintel 的相似之处; Monkaa 电影不追求相同数量的自然性。

我们选择了一些合适的电影场景并额外使用 Monkaa 的片段创建了全新的场景。为了增加数据量，我们渲染多个版本的自制场景，每个都有随机增量更改摄像机的平移和旋转的关键帧。

4.3 Driving

Driving 场景几乎都是从驾驶汽车的角度看去的自然的动态的街道场景，类似于 KITTI 数据集。它使用来自 FlyingThings3D 数据集相同资源池中的汽车模型，并额外使用来自 3D Warehouse 的高度详细的树模型和简单的路灯。在图 4 中，我们展示了从 Driving 中选定的帧以及 KITTI 2015 中相似的帧。

我们的立体视觉基线设置为 1 个 Blender 单元，还有典型的车型宽度约 2 个单位相当于 KITTI 的设置（54 厘米基线，186 厘米车宽[8]）。

图 4: 来自 KITTI 2015 基准套件[17] 和我们的新的 Driving 数据集的示例帧。两个都显示了许多从各种现实的视角的静态和移动汽车，稀薄的对象，复杂的阴影，纹理化的地面，具有挑战性的镜面反射。

5．网络

为了证明我们新的合成数据集在场景流估计中的适用性，我们使用它来训练卷积网络。一般来说，我们遵循 FlowNet[4]架构。也就是说，每个网络包含收缩部分和扩展部分以及两部分之间的远程链路。收缩部分包含步长为偶尔使用的 2 的卷积层，导致总的下采样因子为 64。这允许网络来估计大位移。网络的扩展部分接下来逐步和非线性地上采样特征图，同时考虑到来自收缩部分的特征。这是通过一系列上卷积层和卷积层实现。 注意，网络中没有数据瓶颈，信息也可以通过收缩层和扩展层之间的远程链路。整体架构的说明我们可以参考论文 Dosovitskiy 等人[4]的图。

对于视差估计，我们建议使用基本的 DispNet 架构，如表 2 所示。我们发现在扩展部分中的额外卷积产生比 FlowNet 架构更平滑的视差图; 见图 6。

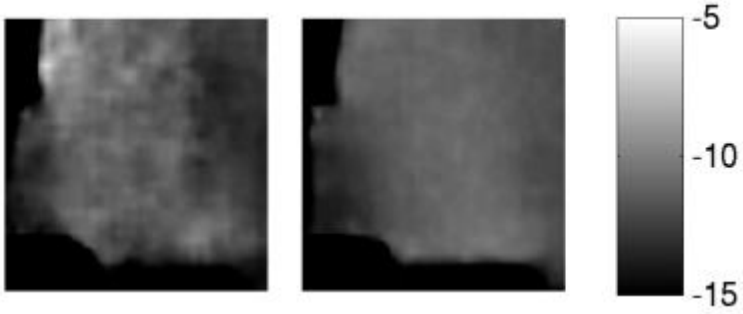

图 6: 上卷积之间没有（左图）和有（右图）额外卷积的预测视差图的特写。注意右图的预测更平滑。

表 2: DispNet 架构的说明。收缩部分由卷积 conv1 至 conv6b 组成。在扩展部分，上卷积（upconvN），卷积（iconvN，prN）和损失层是交替的。来自早期层的特征连接着更高层的特征。预测视差图由 pr1 输出。

Name	Kernel	Str.	Ch I/O	InpRes	OutRes	Input
conv1	7×7	2	6/64	768×384	384×192	Images
conv2	5×5	2	64/128	384×192	192×96	conv1
conv3a	5×5	2	128/256	192×96	96×48	conv2
conv3b	3×3	1	256/256	96×48	96×48	conv3a
conv4a	3×3	2	256/512	96×48	48×24	conv3b
conv4b	3×3	1	512/512	48×24	48×24	conv4a
conv5a	3×3	2	512/512	48×24	24×12	conv4b
conv5b	3×3	1	512/512	24×12	24×12	conv5a
conv6a	3×3	2	512/1024	24×12	12×6	conv5b
conv6b	3×3	1	1024/1024	12×6	12×6	conv6a
pr6+loss6	3×3	1	1024/1	12×6	12×6	conv6b
upconv5	4×4	2	1024/512	12×6	24×12	conv6b
iconv5	3×3	1	1025/512	24×12	24×12	upconv5+pr6+conv5b
pr5+loss5	3×3	1	512/1	24×12	24×12	iconv5
upconv4	4×4	2	512/256	24×12	48×24	iconv5
iconv4	3×3	1	769/256	48×24	48×24	upconv4+pr5+conv4b
pr4+loss4	3×3	1	256/1	48×24	48×24	iconv4
upconv3	4×4	2	256/128	48×24	96×48	iconv4
iconv3	3×3	1	385/128	96×48	96×48	upconv3+pr4+conv3b
pr3+loss3	3×3	1	128/1	96×48	96×48	iconv3
upconv2	4×4	2	128/64	96×48	192×96	iconv3
iconv2	3×3	1	193/64	192×96	192×96	upconv2+pr3+conv2
pr2+loss2	3×3	1	64/1	192×96	192×96	iconv2
upconv1	4×4	2	64/32	192×96	384×192	iconv2
iconv1	3×3	1	97/32	384×192	384×192	upconv1+pr2+conv1
pr1+loss1	3×3	1	32/1	384×192	384×192	iconv1

我们还测试了一个利用显式相关层[4]的架构，我们称之为 DispNetCorr。在这个网络中，两个图像单独处理，直到 conv2 层，然后将所得到的特征水平相关。我们认为最大位移为 40 像素，其对应于输入图像中的 160 个像素。与 Dosovitskiy 等人[4]的 2D 相关相比，1D 相关在计算上更简洁并且允许我们用比 FlowNet 更精细的采样来覆盖更大的位移，对这个相关，我们使用步长为 2。

我们训练一个用于场景流估计的联合网络，它组合并且微调预训练的视差和光流网络。如图 5 所示。我们使用我们的 FlowNet 实现方法来预测左右图之间的光流和两个 DispNets 来预测 t 和 t+1 图的视差。 然后，我们微调这个大组合网来估计光流，视差和额外的视差变化。

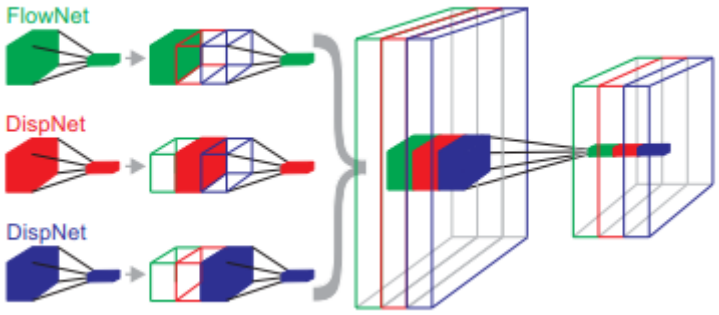

图 5: 交叉 FlowNet（绿色）和两个 DispNets（红色和蓝色）的权重到 SceneFlowNet。对于每一层，过滤器掩码是通过设置一个网络（左）的权重为 1 并将其他网络的权重分别设为零（中间）。然后将来自每个网络的输出连接产生一个大网络，这个网络具有三倍的输入数和输出数（右）。

训练。所有网络都是端到端的训练，给定图像作为输入和地面实况（光流，视差，或场景流）作为输出。我们使用自定义版本的 Caffe[13]和利用 Adam optimizer 优化器[14]。我们设置 β1=0.9，β2=0.999 如论文 Kingma 等人[14]。同时我们使用的学习率 λ=1e-4 并且从迭代 400000 次开始的每 200000 次迭代学习率除以 2。

由于网络的深度以及收缩层和扩展层之间的直接连接（见表 2），如果所有的六个损失都是激活的，更低的层得到混合梯度。我们发现使用损失权重法是有效的：我们开始训练时，将损失权重 1 分配给最低分辨率损失 loss6，将权重 0 设为所有其他损失（即所有其他损失均被关闭）。在训练期间，我们逐步增加具有更高分辨率的损失的权重，停用低分辨率的损失。这使得网络能够首先学习粗略表示然后继续更精细的分辨率的处理，没有损失限制中间特征。

数据扩增。尽管有大量的训练集，我们选择执行数据扩增以几乎没有额外代价的引入更多的种类到训练数据中。我们执行空间变换（旋转，平移，剪切，缩放）和色彩变换（颜色，对比度，亮度），我们对所有 2 或 4 输入的图像使用相同的变换。

对于视差，引入任何旋转或垂直偏移都会打破对极约束。水平移位会导致负视差或者移至相机的无穷远。

6. 实验

现有方法的评估。我们用我们的数据集评估了几个现有的视差和光流估计方法。对于差异，我们评估 Zbontar 和 LeCun 的最先进的方法[28]和流行的半全局匹配[11]方法与 OpenCV 的块匹配方法。结果与那些我们的 DispNets 一起展示在表 3 中。我们使用端点误差（EPE）作为大多数情况的误差测量，唯一的例外是 KITTI 2015 测试集只有 D1-all 误差是由 KITTI 的评估服务器提供的。它是估计误差比地面实况视差大 3px 和大 5%的像素的百分比。

表 3: 视差误差。所有测量值都是端点误差，除了 KITTI-2015 测试的 D1-all 测量值（参见文本解释）。它的结果来自 KITTI 2012 训练的微调网络。

Method	KITTI 2012		KITTI 2015		Driving	FlyingThings3D	Monkaa	Sintel Clean	Time
	train	test	train	test (D1)		test		train	
DispNet	2.38	—	2.19	—	15.62	2.02	5.99	5.38	0.06s
DispNetCorr1D	1.75	—	1.59	—	16.12	1.68	5.78	5.66	0.06s
DispNet-K	1.77	—	(0.77)	—	19.67	7.14	14.09	21.29	0.06s
DispNetCorr1D-K	1.48	1.0†	(0.68)	4.34%	20.40	7.46	14.93	21.88	0.06s
SGM	10.06	—	7.21	10.86%	40.19	8.70	20.16	19.62	1.1s
MC-CNN-fst	—	—	—	4.62%	19.58	4.09	6.71	11.94	0.8s
MC-CNN-acrt	—	0.9	—	3.89%	—	—	—	—	67s

DispNet。我们在 FlightThings3D 数据集上训练 DispNets 网络，然后在 KITTI 上对其进行选择性的微调。微调后的网络在表中用“-K”后缀表示。DispNetCorr 是在 KITTI 2015 上微调的，是目前在 KITTI 2015 年前几名表中的第二名，略低于 MC-CNN-acrt[28]，MC-CNN-acrt 大约快 1000 倍。关于 KITTI 分辨率，它以每秒 15 帧的速度运行在 Nvidia GTX TitanX GPU 上。对于前景像素(属于汽车模型)它的误差大约是文献[28]的一半。该网络误差比表中最好的实时方法 Multi-Block-Matching[7]还要低 30%。在其他数据集中 DispNet 也表现良好，并且优于 SGM 和 MC-CNN 网络。

虽然 KITTI 的微调改善了在这个数据集中的结果，但这也会增加在其他数据集中的误差。我们认为这个显著的性能下降，是由于 KITTI 2015 中只包含相对较小的是视差，最高约 150 像素，而其他数据集包含的一些 500 像素及以上的视差。当在 KITTI 上进行微调时，网络似乎失去了预测大位移的能力，因此在其他数据集中导致了巨大的错误。

对比着 FlowNet[4]，我们对网络架构提出了几个修改。首先，我们在网络扩

展部分的上卷积层之间的添加了一层卷积层。正如预期，这允许网络更好地调整视差图并预测出更平滑的结果，如在图 6 所示。在数量上，这导致在 KITTI 2015 中相对 EPE 评估大约降低了 15%。

第二，我们使用一个 1D 相关层训练了我们网络的一个版本。与 Dosovitskiy 等人[4]相比，我们发现具有相关性的网络整体更好，（见表 3）。一个可能的解释是，1D 性质的视差估计问题允许我们在比 FlowNet 更细的网格上计算相关性。

SceneFlowNet。我们提供用卷积网络场景流估计的完整的早期结果。图 8 显示了一个 FlyingThings 场景的网络的结果。这个网络能够很好地预测视差变化，即使在被遮挡的区域。由于训练场景流时必须处理的数据量很大，网络训练相对较慢（一个前向传递的网络需要 0.28s，是 DispNet 上时间的 5 倍），而且还没有收敛。随着我们允许网络训练更长时间，我们希望结果进一步改善。表 4 是我们数据集的定量评估

表 4: 在上述数据集中评估我们的 SceneFlowNet 的端点误差。Driving 数据集包含最大的视差，光流和视差变化，导致了较大的误差。FlyingThings3D 数据集包含大光流，而 Monkaa 包含较小的光流和较大的视差。

SceneFlowNet	Driving	FlyingThings3D	Monkaa
Flow	22.01	13.45	7.68
Disparity	17.56	2.37	6.16
Disp. change	16.89	0.91	0.81

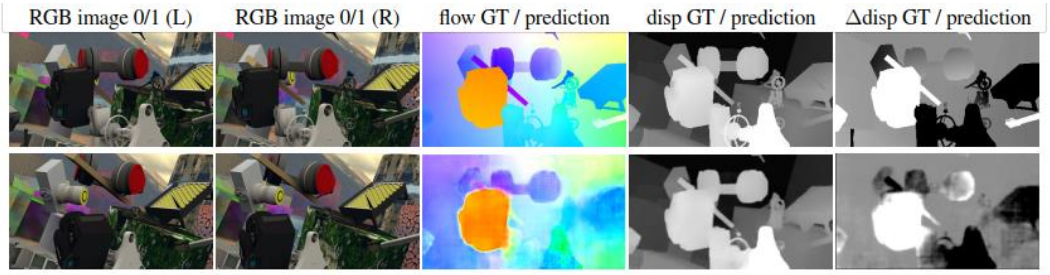

图 8: 预训练的 FlowNet 和 DispNets 产生的 SceneFlowNet 结果。已经加入视差变化并在 FlyingThings3D 的 80000 次迭代中进行了网络微调。视差变化的预测在经过这几次训练的迭代已经相当不错了。

7. 总结

我们已经介绍了一个包含超过 35000 个带地面实况视差，光流和场景流的立体图像对的合成数据集。虽然我们的动机是创造一个足够大的适合于训练卷积网络估计的数据集，但这个数据集也可以用于评价其他方法。特别是对场景流来说

很有趣，那里一直缺乏带地面实况信息的数据集。

我们已经证明，这个数据集确实可以成功地训练大型卷积网络：我们训练的视差估计的网络是和当前最先进的网络同等水平的，当前最先进的网络比我们的运行速度快 1000 倍。率先使用标准的网络架构训练场景流估计网络的方法也显示出了不错的效果。我们相信我们的数据集将有助于提高深入学习研究中立体视觉，光流和场景流估计这样具有挑战性的视觉任务。

8．致谢

这项工作的一部分资金来自 ERC Starting Grant VideoLearn，the ERC Consolidator Grant 3D Reloaded，DFG Grants BR3815 / 7-1 和 CR 250 / 13-1。

参考文献

[1] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski. A database and evaluation methodology for optical flow. Technical Report MSR-TR-2009-179, December 2009. 2

[2] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In ECCV, Part IV, LNCS 7577, pages 611–625, Oct. 2012. 2

[3] J. Cech, J. Sanchez-Riera, and R. P. Horaud. Scene flow estimation by growing correspondence seeds. In CVPR, 2011.3

[4] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazırbas¸, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In ICCV, 2015. 1, 2, 3, 5, 7, 12

[5] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In CVPR, 2015. 3

[6] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. NIPS, 2014. 3

[7] N. Einecke and J. Eggert. A multi-block-matching approach for stereo. In

Intelligent Vehicles Symposium, pages 585– 592, 2015. 7

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. International Journal of Robotics Research (IJRR), 2013. 2, 5

[9] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge university press, 2003. 12

[10] J. Hays and A. A. Efros. im2gps: estimating geographic information from a single image. In CVPR, 2008. 4

[11] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. PAMI, 30(2):328–341, 2008. 3, 6

[12] F. Huguet and F. Deverney. A variational method for scene flow estimation from stereo sequences. In ICCV, 2007. 3

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Gir-shick, S. Guadarrama, and T. Darrell. Caffe: Convolu-tional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093, 2014. 5

[14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In ICLR, 2015. 5

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, pages 1106–1114, 2012. 2

[16] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4):541–551, 1989. 2

[17] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In Conference on Computer Vision and Pattern Recognition (CVPR), 2015. 2, 3, 5

[18] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In ECCV, 2012. 2

[19] J. Quiroga, F. Devernay, and J. Crowley. Scene flow by tracking in intensity and depth data. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, pages 50–57. IEEE, 2012. 3

[20] M. Savva, A. X. Chang, and P. Hanrahan. Semantically-Enriched 3D Models for Common-sense Knowledge. CVPR 2015 Workshop on Functionality, Physics,

Intentionality and Causality, 2015. 4

[21] D. Scharstein, H. Hirschmuller, Y. Kitajima, G. Krathwohl, N. Neˇc, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In Pattern Recognition, pages 31–42. Springer, 2014. 2

[22] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International journal of computer vision, 47(1-3):7–42, 2002.2

[23] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(3):475–480, 2005. 3

[24] S. Vedula, S. Baker, P. Rander, R. T. Collins, and T. Kanade. Three-dimensional scene flow. In ICCV, pages 722–729, 1999. 3

[25] C. Vogel, K. Schindler, and S. Roth. Piecewise rigid scene flow. In ICCV, 2013. 3

[26] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, and D. Cremers. Efficient dense scene flow from sparse or dense stereo data. Springer, 2008. 3

[27] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In Computer Vision (ICCV), 2013 IEEE International Conference on, pages 1625–1632, Dec 2013. 2

[28] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. arXiv preprint arXiv:1510.05970, 2015. 3, 6, 7

[29] K. Zhang, J. Lu, and G. Lafruit. Cross-based local stereo matching using orthogonal integral images. IEEE Trans. Circuits Syst. Video Techn., 19(7):1073–1079, 2009. 3

一个训练视差，光流，场景流估计卷积网络的大型数据集：补充材料

1．介绍

由于本文篇幅的限制，补充材料包含了数据集产生过程更详细的描述（第 2 节），以及更多的细节和 DispNet 更多的定性结果（第 3 节）。

图 1: Driving 场景的鸟瞰图。相机沿着街道上的一条羊肠小道，并遇到很多转角，道口，其他车辆和不同的照明条件。

2．数据集产生细节

我们修改 Blender 的的通道的内部渲染引擎来产生——除了立体 RGB 图像——每帧每视图三个额外的数据传递。图 2 给出了这个数据的可视明细：

- 在基础传递中（3DPos$_t$），每个像素存储现场点的真实 3D 位置，它投影那个像素（3D 位置在相机坐标系统中给出）。
- 对于第二次传递（3DPos$_{t-1}$），我们恢复时间到前一帧 t-1，并保存那个时候的所有顶点的 3D 位置。然后我们返回到当前帧 t 并使用时间 t 的顶点 3D 位置来投影时间 t-1 的 3D 顶点到图像空间中。因此，我们再次存储每个像素的 3D 位置，但此时从时间 t-1 的 3D 位置，使用在时间 t 的投影。
- 第三次传递（3DPos$_{t+1}$）类似第二次传递，但这次我们使用的后续的 t + 1 帧，而不是以前的 t-1 帧。

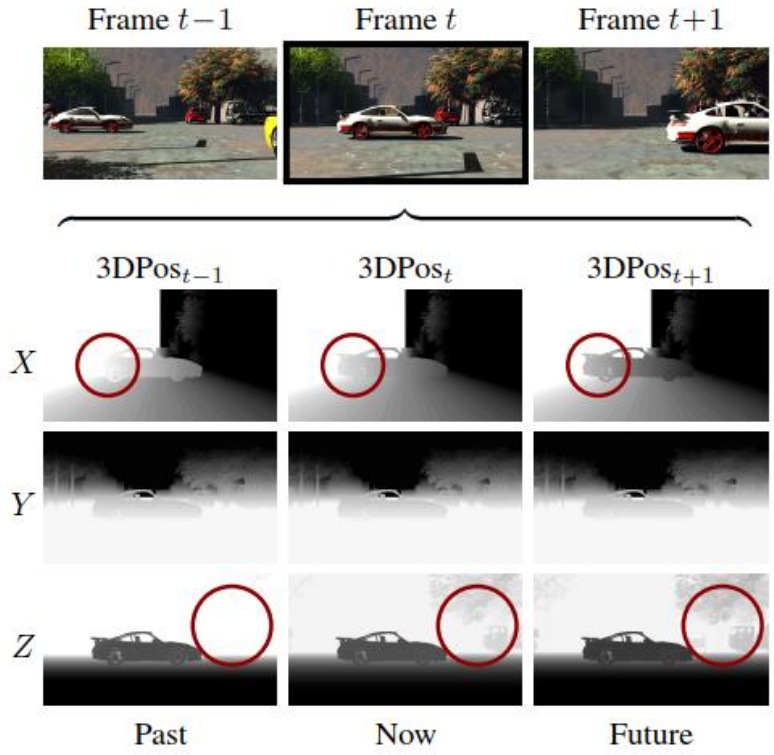

图 2: 我们对 t 帧的中间渲染数据：在 X / Y / Z 通道编码所有帧 t 处可视点（中间列）的 3D 位置（相对于相机），以及他们各自的 3D 位置是/将是在前一帧/下一帧（左/右列）。前一帧和下一帧的 3D 位置是存储在如在帧 t 相同的图像位置。因此，分析来自帧 t 的位置可以得到相应的 3D 点的过去，当前和未来的位置信息。所有场景流数据可以从该信息导出。例如：汽车向右移动改变其 X 值（注意，该透视投影压缩远处天空的强度梯度到一个位于 X= 0 的明显的步骤）。没有什么是垂直运动的，所以所有的 Y 值是随时间恒定的。相机向前移动，所有的 Z 值均匀改变（注意，右侧的对象是如何变得可见的）。

这三个数据结构包含有关的所有关于三维结构和从当前视点可见的场景的三维运动的信息。从 3DPos 数据中，我们生成场景流数据。图 3 描述了从 blender 输出到所得数据集的数据转换步骤。注意彩色图像和分割掩码是直接通过 Blender 产生的，并且不需要任何后期处理。与相机的内在和外在性质一起，各种数据可以产生出来，包括校准的 RGBD 图像。

图 4 展示了对我们数据集中一帧的分割掩码示例。材料可以穿过对象共享，但对象索引和材料索引的组合产生一个场景的独特过度分割（整个场景中的所有帧一致的）。虽然我们的实验不利用这些数据，但对于其它应用，我们还在我们的数据集中包括对象和材质 ID。

与这份补充材料一起，我们还提供一个视频来演示我们创建的数据集和最终

的输出管道，即光流，视差，视差改变，对象和材料索引地面实况。

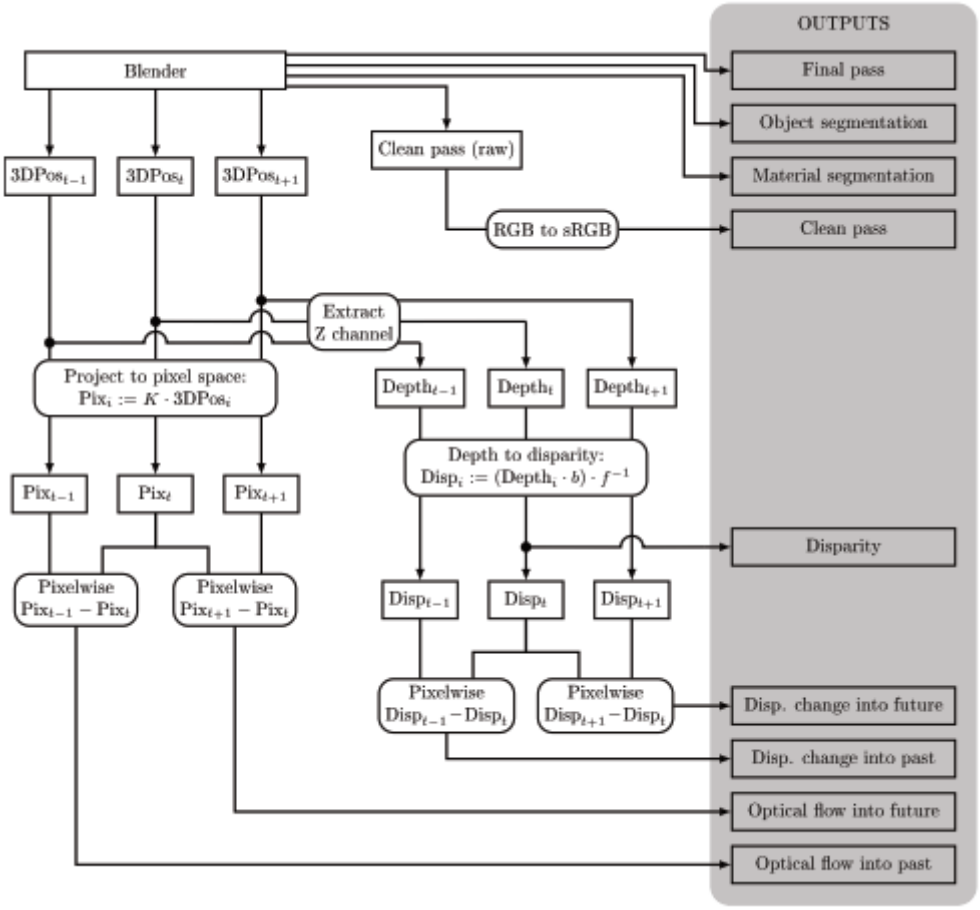

图 3: 在帧时间 t 的单一视图的数据生成概述：Blender 直接输出 Final pass 和 Clean pass 图像，以及对象和材料的分割掩码。视差是直接从深度中获取，深度是由图 2 中的当前 3DPos 图的 Z 通道给出的（b 是立体视觉基线，f 表示焦距）。在未来/过去方向上的视差变化，是未来/过去的视差图的结果减去当前视差图得到的。原始 3DPos 图像从相机空间使用相机内部矩阵 K 转换为像素空间的投影。从未来/过去像素位置图像减去当前的像素位置图像产生了未来/过去的光流。

图 4: 分割数据：对象索引是每个场景独一无二的。材料索引可以跨对象共享，但是可以与对象索引组合以产生过度分割部分。

3．DispNetCorr

直观地看，简单的 DispNet 视差估计架构（如在主论文中所述）必须学习在矫正立体图像中匹配不同的图像部分的概念。由于这个问题的架构是众所周知的（对应只能根据对极几何[9]找到），我们引入另一种可选的架构——DispNetCorr——我们在其中明确沿水平扫描线的相关特性。

虽然 DispNet 使用两个堆叠的 RGB 图像作为单输入（即一个六声道输入的对象），DispNetCorr 架构首先独立处理输入图像，那么关联两个图像之间的特征并进一步处理结果。此行为类似于关联架构[4]中使用的那样，其中 Dosovitskiy 等人用有限相邻大小和每张图不同的步长来构建一个二维相关层。对于视差估计，我们可以用一个没有步长而有较大的邻域大小的更简单的方法，因为沿着一个维度相关性计算上要求更少。我们通过限制仅搜索一个方向，可以额外降低比较的次数。例如，如果我们都给定左摄像机图像，在右图查找对应关系，那么所有的视差位移都是到左边的。

给定两个特征对象 a 和 b，它们有多个信道和相同的大小，我们使用 D 通道计算相同的宽度和高度的关联图，其中 D 是可能的差异值的数量。对于在第一个对象 a 的位置（x，y）的一个像素，在通道 d∈[0，D - 1]的产生的关联入口是两个特征矢量$a_{(x,y)}$和$b_{(x-d,y)}$的数积。

4．定性示例

我们的视差估计网络的定性评估图，并与其他方法的比较，都展示在图 5 到 10 中。

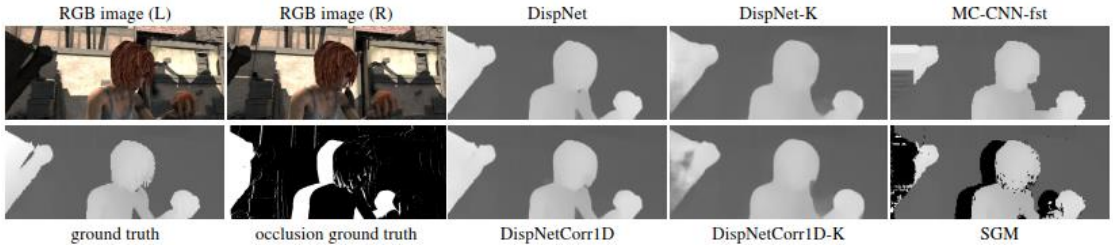

图 5: 一个 Sintel 帧的视差：DispNet 和 DispNetCorr1D 以更合理的方式填充遮挡区域，对比其他方法。

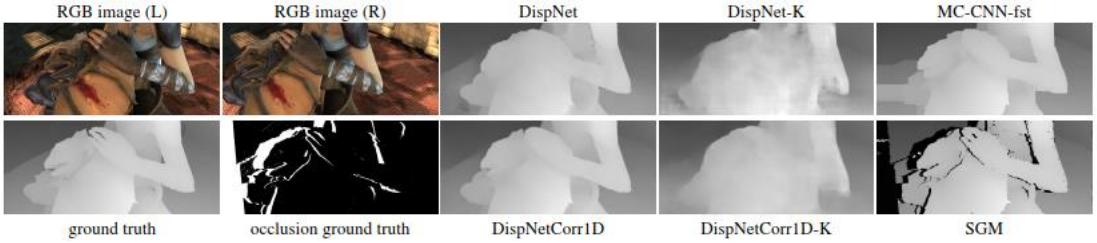

图 6: 一个 Sintel 帧视差：DispNetCorr1D 提供更清晰的估计和对龙首的平滑区域估计比 DispNet 更好。

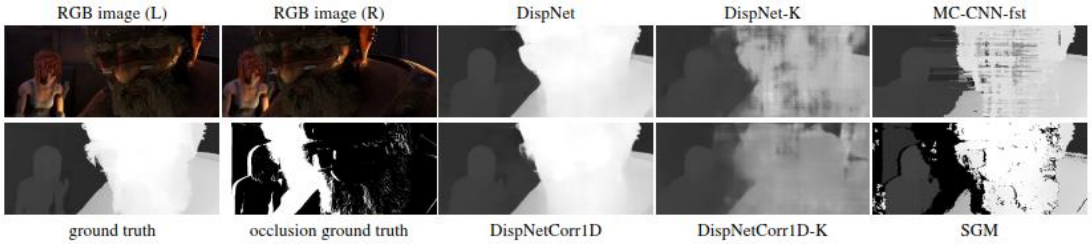

图 7: 一个 Sintel 帧视差：微调网络在 KITTI 2015 数据集中无法估计的巨大视差了（大视差不存在于 KITTI 中）。另外 MC-CNN-FST 也对大视差处理有问题。

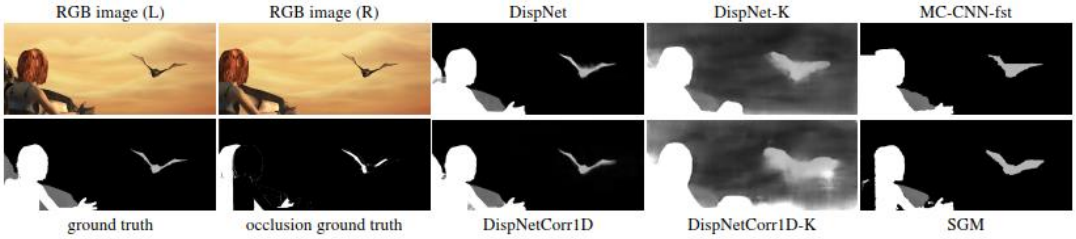

图 8: 一个 Sintel 帧视差：DispNet 和 DispNetCorr1D 可以用一个不错的方式处理遮挡区域。在 KITTI 2015 中微调后，在天空区域失败了（地面对天空真实数据和其他不在 KITTI 中的小视差）。

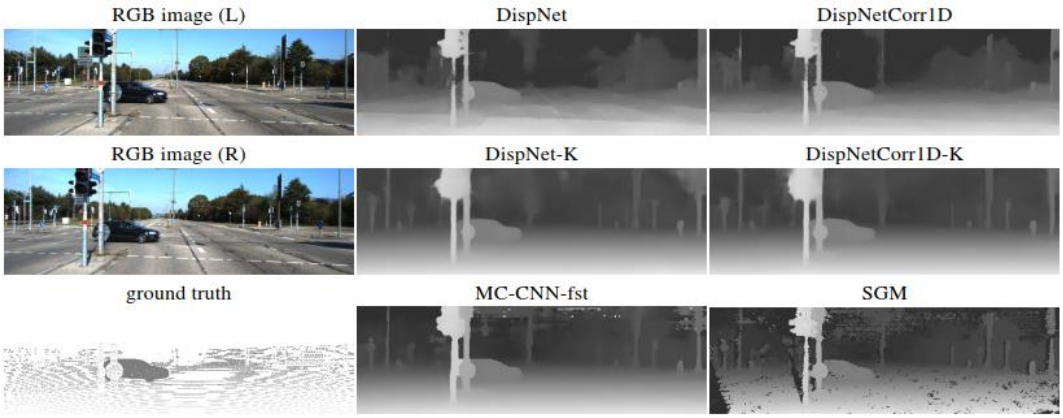

图 9: KITTI 2015 帧的视差：当使用这样的地面实况信息微调时，KITTI 2015 数据集的稀疏性导致了非常平滑的预测。虽然未微调的 DispNet 和

DispNetCorr1D 准确地估计精细细节，但它们对 KITTI 数据集中常见的平滑道路和地面区域估计不太准确。

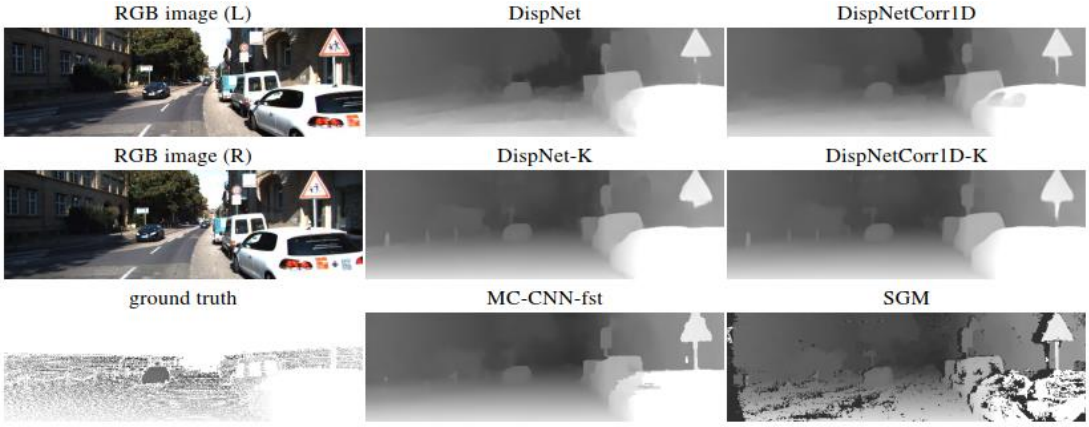

图 10: KITTI 2015 帧的视差：KITTI 的微调网络导致更平滑的估计。然而，DispNet-K 和 DispNetCorr1D-K 仍可以识别在左下角的标识，这是 DispNet 和 DispNetCorr1D 完全忽略的。这表明微调网络不仅仅是过度平滑，但仍能找到小的结构和视差不连续性。