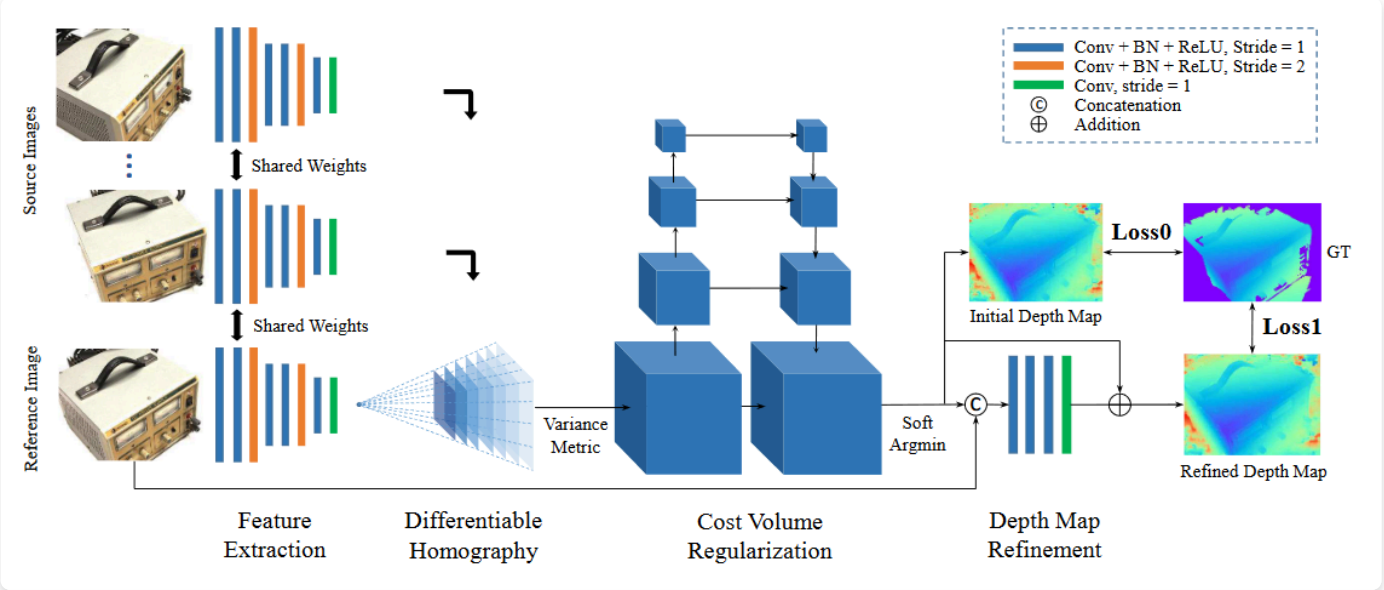
MVSNet

论文：《MVSNet: Depth Inference for Unstructured Multi-view Stereo》
地址：https://arxiv.org/abs/1804.02505
年份：ECCV 2018 (oral)

Introduction

任务：深度图估计
技术贡献：
(1) 引入 CNN 来进行 Multi-view Stereo，使用端到端的深度学习框架来重建深度图；
(2) 提出了可微分的单应变换操作，在相机视锥体中构建了 3d 的 cost volume，以及后续的一些 refine 操作。

Method



MVSNet 的输入是一张 reference image 和 多张 source image，输出是 reference image 对应的深度图。pipeline 如上图所示，首先使用同一个 CNN 对输入图像提取特征，然后根据相机参数将所有 feature map warp 到 reference camera 下的视锥体中，得到 N 个 feature volume。然后对这些 feature volume 计算方差得到一个 3d cost volume，作为 3D U-Net 的输入，网络输出得到深度图，最后经过一个网络对深度图进行 refine，得到最终结果。

Image Features

使用 2D CNN 对输入图像 $\{\mathbf{I}_i\}_{i=1}^N$ 提取特征得到 $\{\mathbf{F}_i\}_{i=1}^N$，输出的是 32 个通道的特征图，且长和宽都 downside 了 4 倍。即 $[N, H, W, 3] \rightarrow [N, H/4, W/4, 32]$

Cost Volume

接下来是构建 3D cost volume 的过程，记 $\mathbf{I}_1$ 为 reference image，$\{\mathbf{I}_i\}_{i=2}^N$ 为 source image，$\{\mathbf{K}_i, \mathbf{R}_i, \mathbf{t}_i\}_{i=1}^N$ 为相机内外参。关于 cost volume 是什么，可以看 https://www.zhihu.com/question/297481800/answer/2248769480。

Differentiable Homography

这里主要涉及两个问题：(1) 给定 2 张图像的相机参数，对于 1 张图像中的某个像素，这个像素对应于 3d 空间中的一个点，如何找到这个点在另一张图像中的位置，也就是找到这两个图像像素之间的对应关系；(2) 有了这样的对应关系，如何构建 cost volume。
对于第一个问题，具体可以看 https://zhuanlan.zhihu.com/p/138266214 的推导。简单来说，对于 reference 坐标系下的像素 $\mathbf{q}$，其对应 3d 空间中的 $\mathbf{p}$，我们想将其转换到 source 坐标系下的像素 $\mathbf{q}'$。转换过程就是一个 2d -> 3d -> 2d 的过程，由于 2d -> 3d 的过程中我们需要知道 $\mathbf{p}$ 的深度 z，因此引入了 $\mathbf{p}$ 所在平面信息 $\{\mathbf{n}, d\}$，最终单应矩阵形式如下：

$$\mathbf{H} = \mathbf{K}'\mathbf{R}(\mathbf{I} - \frac{\mathbf{t}\mathbf{n}^T}{d})\mathbf{K}^{-1}$$

这里的 $\mathbf{R}$ 和 $\mathbf{t}$ 都是两个坐标系之间的相对变换，转换为各自的外参，并用论文中的记号：

$$\mathbf{H} = \mathbf{K}_i\mathbf{R}_i(\mathbf{I} - \frac{(\mathbf{R}_i^{-1}\mathbf{t}_i - \mathbf{R}_1^{-1}\mathbf{t}_1)\mathbf{n}_1^T\mathbf{R}_1}{d})\mathbf{R}_1^{-1}\mathbf{K}_1^{-1}$$

可以发现论文中给出的公式是完全错误的。
这样的话，对于 reference image 中的每一个像素，都可以根据公式计算出其在 source images 下对应的像素坐标。但是公式中还需要 3d 点的 d (等价于我们希望求的深度)，因此参考了 plane sweeping stereo 的思想 (介绍可见 https://www.codetd.com/article/2992701)，我们需要的 d 在某个范围内 (如 near 和 far 平面之间)，可以在这个范围内枚举 D 个 d 的值，对每个像素都根据这些 d 值去找 source image 下的对应像素。这样我们相当于得到了 D 种从 reference image 坐标到 source image 坐标的坐标转换方式，再查坐标对应的 source image 的值，就得到 D 张图像，且其排列在 reference 相机的视锥体中，成为 feature volume。可以想象得到，D 张图像中的每张图像都会只有部分区域是准确的 (和 reference image 是同一个值)，交给后续步骤去处理得到一张深度图。
简单总结一下，转换过程是一个 2d->3d->2d 的过程，只是 3d 这里是不确定的，每个像素对应了视锥体中的一条线，所以 3d -> 2d 后就有多个可能的值，把这些可能的值都算了出来，结果就相当于在视锥体中的多张图片，后续就要根据某种规则去挑选这些图片中准确的区域，整合得到最终的结果。

Cost Metric

经过上一步的单应变换，我们得到了多个 feature volume $\{\mathbf{V}_i\}_{i=1}^N$，现在要将其整合为一个 cost volume $\mathbf{C}$，这一步相当于整合多个视角下的信息，用于后续准确估计深度图。
采用计算方差的方式计算 cost volume：

$$\mathbf{C} = \mathcal{M}(\mathbf{V}_1, \cdots, \mathbf{V}_N) = \frac{\sum_{i=1}^N (\mathbf{V}_i - \bar{\mathbf{V}}_i)^2}{N}$$

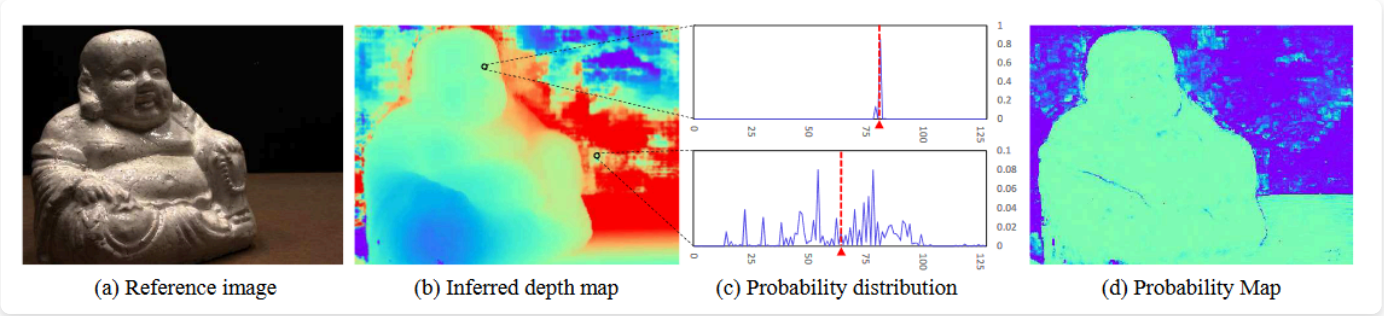
我们知道如果猜测 d 是准确的话，那么在多个视角下这个位置的值都会是相同的，也就是计算的方差接近于 0。所以挑选那些方差为 0 的区域对应的深度值，就可以构建出大致的深度图了。但结果可能还会存在部分噪声，所以就需要后面的方法去继续改进。

Cost Volume Regularization

由于存在 non-Lambertian 表面以及物体遮挡的情况，通过上面方法计算得到的 cost volume 还是包含噪声的，所以想要通过 regularization 来 refine cost volume $\mathbf{C}$ 得到 probability volume $\mathbf{P}$ 来进行深度估计。
这里采用 U-Net 架构的 3D CNN，将 32 通道的 cost volume 整合为 1 通道的 volume，最后使用 softmax 来归一化得到概率。最终的输出就是一个 probability volume，每个像素坐标都有一个关于深度的概率分布。这样做有两个**好处**：(1) 能够很容易估计深度，概率密度最大的 d 就可以作为该点的深度值；(2) 还可以做置信度的估计，看估计得到的深度是否可靠，如果是可靠的深度，那么概率分布会类似一个单峰分布，只在某点概率高，其他点概率低，而不可靠的深度通常会是一个“多峰”分布，因此可以作为 outlier 剔除。

Depth Map

我们首先根据 probability volume 估计深度图，然后剔除一些 outlier。



一种估计深度的简单方法就是对概率分布取 argmax，这样就可以得到概率最大的深度值。文章认为这样做有两个不足：(1) "unable to produce sub-pixel estimation"，感觉是 argmax 只能得到离散的整数值，不能得到更精确的带小数的值了；(2) argmax 是不可微的，所以不能进行反向传播。
所以就采用了对 argmax 的近似操作：$\mathrm{soft\ argmin}$，具体就是求一个加权和：

$$\mathbf{D} = \sum_{d=d_{min}}^{d_{max}} d \times \mathbf{P}(d)$$

计算结果可以参照上图中 (c) 处的红线。
我们还可以根据概率分布去评价估计的深度的可靠程度，比较可靠的深度一般是单峰分布，不可靠的一般是“多峰”分布，文章采用的做法是在 $\mathrm{soft\ argmin}$ 算出的 d 值的一个小邻域求概率密度和 (4 个最近的深度值)，小于阈值就认为是 outlier 剔除。其他方法例如标准差和熵都可以用于提出 outlier，但文章通过实验发现其他方法并没有显著的提升。

Depth Map Refinement

现在我们得到一张深度图，文章认为由于在 regularization 过程中使用了较大的感受野，导致深度图中边界区域会有 oversmooth 的问题。但原始的 reference image 中包含了边界等信息，所以就用 reference image 为指导去 refine 深度图，具体做法就是将 reference image 缩小为 $1/4$，与深度图作拼接输入到网络中去学 depth residual，然后将网络输出加到原始深度图中得到最终的深度图。

Experiments

Training

Data Preparation

需要为训练准备 ground truth 的深度图， DTU 数据集只提供了点云和法向，所以通过 screened Poisson surface reconstruction 生成 mesh 表面，然后在各个视角下渲染 mesh 得到深度图。

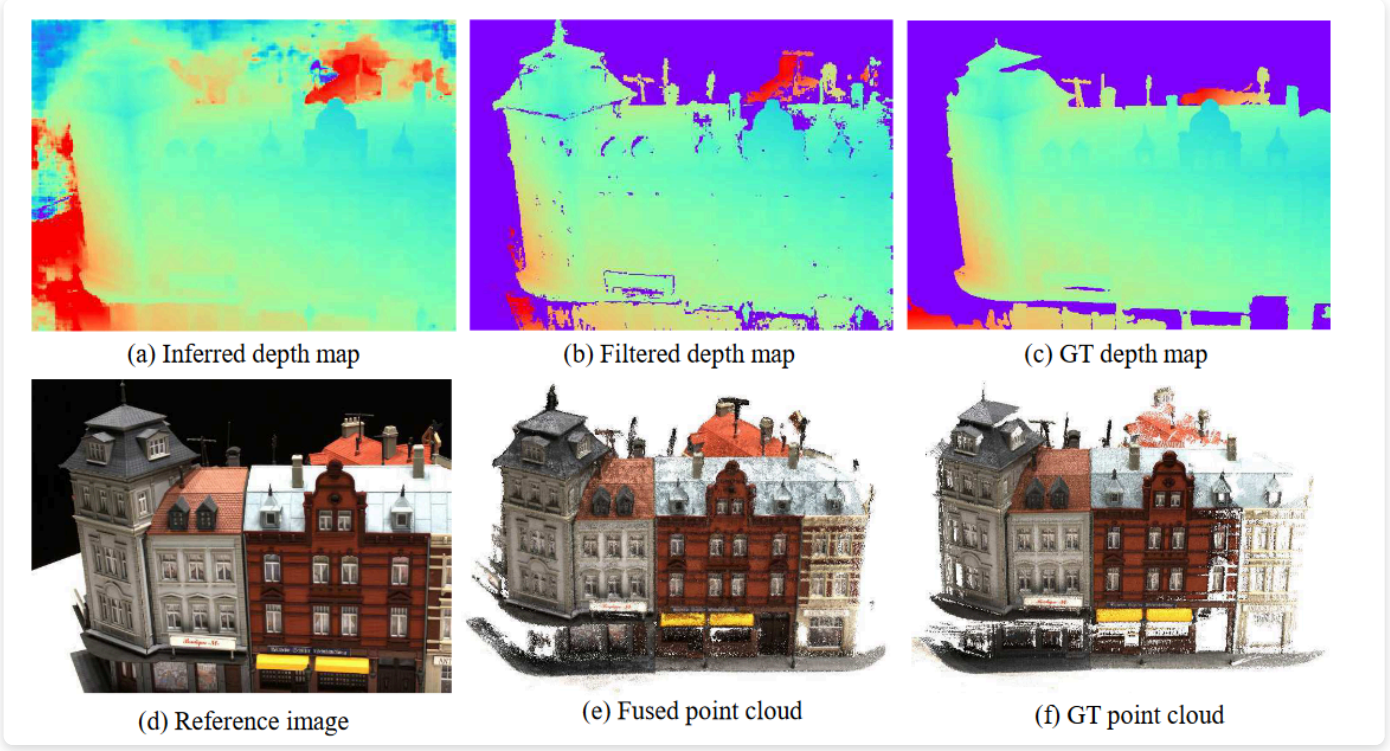
View Selection

训练需要 1 张 reference image 和 2 张 source image，因此需要选取合适的视角作为 source image，通过计算一个分数 $s(i,j)=\sum_{\mathbf{p}}\mathcal{G}(\theta_{ij}(\mathbf{p}))$ 来作为选取标准。其中 $\mathbf{p}$ 是 3d 点，$\theta_{ij}(\mathbf{p})=(180/\pi)\arccos((\mathbf{c}_i-\mathbf{p})\cdot(\mathbf{c}_j-\mathbf{p}))$ 是两个视角形成的 baseline 角，$\mathcal{G}$ 是 piecewise Gaussian function，具体形式可以看原文。

Post-processing

Depth Map Filter

通过网络得到了深度图，想要进一步得到稠密点云，需要去除掉位于背景和被遮挡区域的 outlier，文章提出 photometric and geometric consistencies 来做 filter。
photometric consistency 就是之前提到的用于评价深度可靠性的指标，将小于 0.8 的点当作外点。
geometric consistencies 考虑深度在多个视角下的一致性，对于 reference pixel p_1，根据估计的深度 d_1 将其投影到另一个视角下 p_i，然后再根据另一个视角下估计的深度 d_i，将其重投影回 reference 坐标系下 p_{reproj}，p_{reproj} 也对应一个深度 d_{reproj}，如果 $|p_{reproj}-p_1|<1$ 且 $|d_{reproj}-d_1|/d_1<0.01$，就认为 p_1 估计的 d_1 是两视角一致的。在实验中要求所有深度值至少是三视角一致的。



(a) Inferred depth map (b) Filtered depth map (c) GT depth map

(d) Reference image (e) Fused point cloud (f) GT point cloud

具体的实验结果可以看原论文。
训练的模型也有泛化能力，在 DTU 数据集上训练的模型能够直接用于 TnT 数据集。

Ablations

View Number：训练时使用的视角数越多，效果越好。如果训练时 N 为 3，测试时使用 N 为 5，也能比测试时使用 N 为 3 的效果好。
Image Features：通过网络提取的特征能够帮助提升 MVS 的重建质量。
Cost Metric：通过基于方差计算的 cost volume 相比计算均值的 volume 能够更快地收敛。