基于元学习的双目深度估计在线适应算法

张振宇 [1, 2, 3] 杨 健 [1, 2, 3]

摘 要 双目深度估计的在线适应是一个有挑战性的问题, 其要求模型能够在不断变化的目标场景中在线连续地自我调整并适应于当前环境. 为处理该问题, 提出一种新的在线元学习适应算法 (Online meta-learning model with adaptation, OMLA), 其贡献主要体现在两方面: 首先引入在线特征对齐方法处理目标域和源域特征的分布偏差, 以减少数据域转移的影响; 然后利用在线元学习方法调整特征对齐过程和网络权重, 使模型实现快速收敛. 此外, 提出一种新的基于元学习的预训练方法, 以获得适用于在线学习场景的深度网络参数. 相关实验分析表明, OMLA 和元学习预训练算法均能帮助模型快速适应于新场景, 在 KITTI 数据集上的实验对比表明, 本文方法的效果超越了当前最佳的在线适应算法, 接近甚至优于在目标域离线训练的理想模型.

关键词 深度估计, 在线学习, 元学习, 域适应算法, 深度神经网络

引用格式 张振宇, 杨健. 基于元学习的双目深度估计在线适应算法. 自动化学报, 2023, **49**(7): 1446−1455

DOI 10.16383/j.aas.c200286

Online Adaptation Through Meta-learning for Stereo Depth Estimation

ZHANG Zhen-Yu[1, 2, 3] YANG Jian[1, 2, 3]

Abstract This work tackles the problem of online adaptation for stereo depth estimation, that consists in continuously adapting a deep network to a target video recorded in an environment different from that of the source training set. To address this problem, we propose a novel online meta-learning model with adaptation (OMLA). Our proposal is based on two main contributions. First, to reduce the domain-shift between source and target feature distributions we introduce an online feature alignment procedure derived from batch normalization. Second, we devise a meta-learning approach that exploits feature alignment for faster convergence in an online learning setting. Additionally, we propose a meta-pre-training algorithm in order to obtain initial network weights on the source dataset which facilitate adaptation on future data streams. Experimentally, we show that both OMLA and meta-pre-training help the model to adapt faster to a new environment. Our proposal is evaluated on the KITTI dataset, where we show that our method outperforms both algorithms trained on the target data in an offline setting and state-of-the-art adaptation methods.

Key words Depth estimation, online learning, meta-learning, domain adaptation, deep neural network

Citation Zhang Zhen-Yu, Yang Jian. Online adaptation through meta-learning for stereo depth estimation. *Acta Automatica Sinica*, 2023, **49**(7): 1446−1455

深度估计是视觉场景理解中的基础性问题, 并且越来越多地受到计算机视觉和机器人研究领域的关注. 近年来, 一些基于深度学习的 RGB 自动深度估计方法陆续提出, 取得了令人印象深刻的效果[1−7]. 在这些研究工作中, 深度神经网络模型的训练需要依赖深度真值作为监督信息, 可供训练的深度真值越多则效果越好. 然而, 在现实场景中进行数据收集需要对应的硬件平台和设备 (例如汽车和雷达), 且使其在不同环境中工作相当长的时间. 因此, 数据收集通常需要昂贵的财力和时间开销, 这制约了以上监督学习方法的实际应用. 为了避免开销较大的数据收集和人工标注过程, 一些自监督 (也称为无监督) 深度估计方法相继提出[8−11]. 值得一提的是, 尽管没有精确的深度真值, 这些方法仍能通过图像重构误差训练模型, 获得与监督学习方法接近的结果.

尽管上文提到的自监督方法获得了相当好的结果, 其在现实场景中的使用仍然受到制约. 原因在于这些方法均在封闭世界假设下进行设计和评估,

收稿日期 2020-05-07 录用日期 2020-09-14
Manuscript received May 7, 2020; accepted September 14, 2020
国家自然科学基金 (U1713208) 资助
Supported by National Natural Science Foundation of China (U1713208)
本文责任编委 徐昕
Recommended by Associate Editor XU Xin
1. 高维信息与感知教育部重点实验室 南京 210094 2. 江苏省社会安全图像与视频理解重点实验室 南京 210094 3. 南京理工大学计算机科学与工程学院 PCA 实验室 南京 210094
1. Key Laboratory of Intelligent Perception and Systems for High-dimensional Information of Ministry of Education, Nanjing 210094 2. Jiangsu Key Laboratory of Image and Video Understanding for Social Security, Nanjing 210094 3. PCA Laboratory, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094

这意味着训练和测试数据处于同一个数据集, 或者二者的环境表观差异很小. 当模型在全新的场景中工作时, 由于数据域转移的影响, 方法的结果将大打折扣. 因此, 为了增强方法的实际应用效果, 用于深度估计的深度神经网络模型需要考量开放世界的设定, 即可用的视觉数据是在连续变化的环境中被采集的数据流. 以自动驾驶场景为例, 模型需要连续适应于变化的环境 (如城市、乡村和高速公路场景等) 以及光线场景 (如黑夜、黄昏和白昼等). 也就是说, 深度神经网络模型需要具有在线适应的能力.

根据以上的分析和动机, 本文提出了一种基于元学习的双目深度估计方法, 用于需要快速在线适应的开放世界场景. 方法的框架在图 1 中展示. 首先, 一种新的在线元学习适应算法 (Online meta-learning with adaptation, OMLA) 得以提出, 用于模型的快速在线学习. 具体地, 为了处理源域数据 (即训练数据) 和目标视频数据之间的域转移问题, 模型通过调整源域和目标域的批归一化 (Batch normalization, BN) 层统计量, 使域间特征分布对齐. 该方法受启发于文献 [12−13], 本文对其进行改进, 使该方法适用于在线学习场景. 然后, 特征对齐过程与元学习算法相结合, 利用先前时刻的模型反馈选择能够快速适应未来场景的网络超参数. 此外, 本文提出了一种新的基于元学习的预训练方法 (元预训练). 具体地, 在使用 OMLA 算法进行一段视频的在线适应过程后, 模型评估其在未来帧上的表现, 并以此为反馈更新初始参数. 由此, 模型获得了适用于 OMLA 算法进行快速在线学习的初始参数.

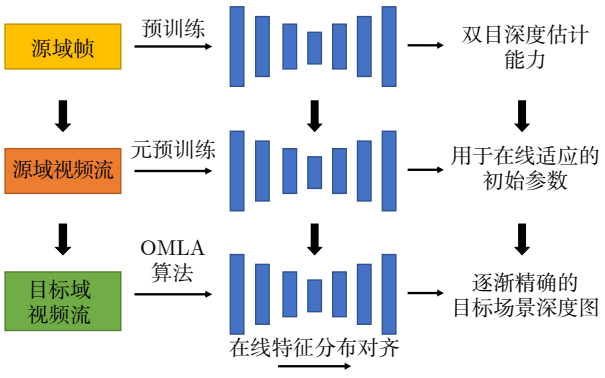

图 1　本文提出的基于元学习的深度估计在线适应算法框架

Fig. 1　The proposed meta-learning framework for online stereo adaptation

本文的主要贡献总结如下: 1) 提出了一种新的基于元学习的在线适应算法, 用于在线视频流的双目深度估计. 该方法与特征对齐过程相辅相成: 元学习算法有助于更新特征对齐动量以提升模型的适应能力; 特征对齐则更好地支持元学习算法进行优化过程的快速收敛. 2) 提出了一种基于元学习的预训练方法, 利用 OMLA 算法使模型获得适用于快速在线学习和收敛的参数以及特征对齐动量. 3) 在 KITTI 数据集[14] 上的实验表明, 本文提出的 OMLA 算法与预训练方法均有助于在线深度估计效果的提升, 其效果超越了当前最佳的在线深度估计适应算法[15].

1　相关工作

1.1　深度估计

随着深度卷积神经网络的发展, 近年来涌现了许多基于监督学习的深度估计方法[1−4, 16−17], 这些方法需要依赖深度真值训练网络模型. 一些大规模数据集, 例如 NYU[18]、KITTI[14, 19] 等为深度估计任务提供了支持, 一些合成数据库的提出 (例如 Synthia[20]) 使得深度估计的迁移学习成为可能.

为了避免耗费较高的数据收集和标注过程, 自监督的深度估计方法[8−10] 得以陆续提出. 例如, Godard 等[9] 利用双目一致性损失函数使自监督训练成为可能. 其他相关工作主要包括结合相机位姿估计的方法[21]、利用对抗学习的方法[10, 22]、基于视觉里程计的方法[17] 以及结合图像超分辨的技术[11]. 最新的研究主要关注高效深度估计网络的设计[6] 和雷达点云的补全[7]. 然而, 以上方法仅适用于封闭世界设定下的深度估计, 缺少对开放世界问题的研究.

1.2　域适应问题

域适应是经典的机器学习问题, 近年来逐渐为计算机视觉和机器人领域所关注[23]. 近期基于深度学习的方法主要通过以下策略减轻域转移造成的影响: 分布对齐损失函数[24−25]、对抗生成网络[26−28] 和域对齐层[13, 29−31]. 在机器人领域, 域适应方法主要用于机器人抓握[12, 27]、可行驶区域分割[31] 以及基于目标定位的机器人控制问题[32−33].

与绝大多数先前工作采用离线设定 (即源域与目标域数据在训练过程中是可以全部获取的) 不同的是, 近期一些域适应算法开始研究在线学习设定下的问题, 即目标域数据以序列化形式获得且其分布不断改变, 相关工作如文献 [12, 30−31]. 一些工作研究了针对深度估计问题的域适应方法[32−35]. Tonioni 等[15] 首先利用在线数据流进行了相关尝试. 文献 [35] 利用元学习方法增强了深度估计模型的在线收敛速度. 然而, 以上两种在线深度估计方法并未清晰地对域转移进行建模, 也未使用任何策略

以减弱源域和目标域数据差别带来的影响. 与其不同的是, 本文方法具体地讨论了在线深度估计中的域转移问题, 并给出了合理且新颖的解决方案.

1.3 元学习算法

元学习的目标是使模型利用先前学习的经验, 学会如何在未来的问题中进行高质量学习. 在文献 [36−39] 中, 元学习被用于获得在新数据域和类别上的快速泛化能力. 在迁移学习应用中, 文献 [38] 利用元学习方法, 使主网络模型引导学生网络合理地微调参数. Park 等在文献 [39] 中提出了一种离线元学习方法, 使网络获得较好的在线跟踪能力. 本文方法受启发于文献 [39], 与其不同的是, 本文方法提出了适用于在线场景的元学习方法, 在引导模型获得合理初值的同时, 使模型在线学习超参数以加速收敛.

2 基于元学习的深度估计在线适应方法

本节详细介绍了本文提出的基于元学习的在线深度估计与适应方法. 为方便形式化讨论, 假定源域 $\mathcal{S}=\{V_s^n\}_{n=1}^N$ 是由 N 个双目视频序列组成的数据集, 这些数据集使用同一个经标定后的双目相机采集. 首先, 使用该源域数据集训练参数为 θ 的神经网络模型 Φ, 使模型获得双目深度估计能力. 然后, 模型需要在目标域视频 $V_{\mathcal{T}}$ 上进行预测, 该视频由不同的双目相机在新的环境中采集, 且视频帧 $I_{\mathcal{T}}^t$ 为数据流形式. 本文方法的目标是在线调整网络参数 θ (即在线适应), 使其在目标域连续变化的视频中逐渐获得更精确的场景深度图.

一种基本的用于在线适应的方法为: 在当前时刻计算当前帧的自监督损失, 并通过梯度下降更新整个网络参数. 尽管这种方法非常简洁, 但其存在明显的缺点, 如其对于域转移是非常敏感的, 且仅关注于当前帧的做法可能会对模型更新过程带来不良影响, 例如场景突变导致的模型漂移会减缓收敛速度. 为了避免这些问题, 本节提出了两种互补策略: 特征分布对齐和在线元学习适应算法 (见图 2). 具体地, 使用批归一化层的统计量对域转移进行建模, 并以此为基础进行源域和目标域的特征分布对齐 (见第 2.1 节). 该过程在前馈中的网络浅层进行, 仅需要非常有限的计算开销. 此外, 使用基于元学习的优化器更新模型, 使模型获得对未来帧的快速收敛能力 (见第 2.2 节). 由此, 在在线适应过程中, 特征对齐能够应对不同域数据的表观特征差异, 元学习优化器则可处理高阶表征的转移问题. 此外, 基于元学习的预训练策略被用于获得模型初始参数 θ_0, 使模型获得针对数据流的快速适应能力 (见第 2.3 节). 训练和更新模型使用的自监督深度估计损失函数在第 2.4 节中介绍.

2.1 在线特征分布对齐的域适应算法

考虑一个包含批归一化层的神经网络模型 Φ, 根据文献 [12−13, 30] 的研究, 域适应可以通过更新 BN 层统计量实现. 在本文问题设定下, 模型无法获得全部目标域数据以计算真实的目标域统计量, 但是由于数据流的存在, 可以随时间渐进地更新统计量使其逼近真实值. 受此启发, 本文提出一种针对 BN 统计量的在线更新策略, 通过目标域数据流逐渐更新 BN 统计量, 使目标域模型特征分布逐渐对齐. 与文献 [13, 30] 的封闭世界方法不同的是, 本文方法无需目标域的全部数据, 适用于数据流可用的开放世界问题. 为简洁起见, 此处仅讨论

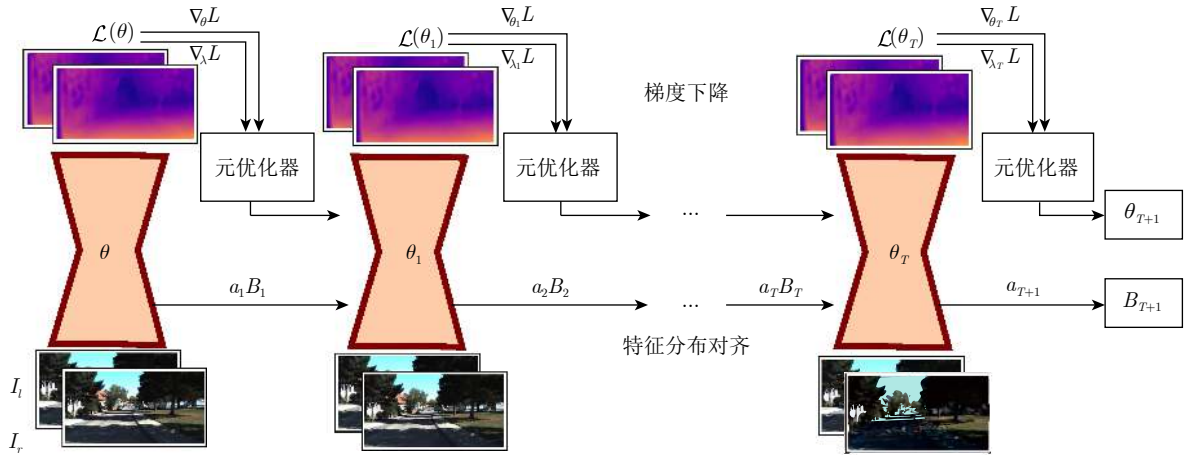

图 2 本文提出的在线元学习适应方法

Fig. 2 The proposed online meta-learning with adaptation (OMLA) method

单个 BN 层的情况, 但相关操作可部署至网络任意 BN 层. 首先, 模型在源域 $\mathcal{S}$ 进行训练后, 可以获得对应的 BN 层统计量 $\mathcal{B}_s = (\mu_s, \sigma_s^2)$. 然后, 模型按照以下过程在目标域视频流中更新统计量. 在 $t = 0$ 时刻, 将统计量初始化 $\mathcal{B}_o = \mathcal{B}_s$. 在 t 时刻, 将先前时刻的 BN 统计量 $\mathcal{B}_{t-1} = (\mu_{t-1}, \sigma_{t-1}^2)$ 对齐至 $\mathcal{B}_t$. 假设特征向量由 m 个样本 $\{x_1, \cdots, x_m\}$ 得到, 则首先计算当前时刻 BN 统计量的观测值

$$\hat{\mu}_t = \frac{1}{m}\sum_{i=1}^{m} x_i \hat{\sigma}_t^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \hat{\mu}_t)^2 \tag{1}$$

给定动量参数 $a_t \in \mathbf{R}$, 以及 $t-1$ 时刻更新后的统计量 $\mathcal{B}_{t-1} = (\mu_{t-1}, \sigma_{t-1}^2)$, 则 t 时刻更新后的统计量为

$$\begin{cases} \mu_t = (1 - a_t)\mu_{t-1} + a_t \hat{\mu}_t \\ \sigma_t^2 = (1 - a_t)\sigma_{t-1}^2 + a_t \dfrac{m}{m-1}\hat{\sigma}_t^2 \end{cases} \tag{2}$$

该更新后的统计量更接近当前场景统计量的真实值. 对该 BN 层给定输入 x, 则此时输出特征为

$$\hat{x} = \gamma \frac{x - \mu_t}{\sqrt{\sigma_t^2 + \epsilon}} + \beta \tag{3}$$

其中, γ 和 β 为 BN 层仿射变换参数, $\epsilon \in \mathbf{R}$ 为极小常量以保持数值稳定. 事实上, 动量 a_t 对特征分布对齐至关重要, 其决定了该 BN 层适应于当前帧的程度. 因此, 与文献 [12] 中手动设定方法不同的是, 本文采用元学习方法自动地获得合适的动量 a_t 值, 使其更好地引导 BN 统计量更新和特征分布对齐. 为简洁起见, 在下文中, a_t 表示所有 BN 层动量的组合.

2.2　在线元学习适应算法

本节介绍所提出的在线元学习适应算法. 假设目标域双目视频序列为 $V = \{I_t\}_{t=0}^{T}$, 其长度为 T, 其中, I_t 为 t 时刻图像对. 当模型在视频 V 上进行在线适应时, 使用算法 1 进行参数更新.

算法 1. 在线元学习适应算法

1. 过程: OMLA ($V = \{I_t\}_{t=0}^{T}$, θ_0, a_0, $\mathcal{B}_0$, λ_0, λ^*)
2. 对 $t = 0$ to T 执行:
3. $\quad D_t, \mathcal{B}_{t+1} = \Phi\left((\theta_t, a_t, \mathcal{B}_t), I_t\right)$
4. $\quad \mathcal{L}_t = \mathcal{L}(D_t, I_t)$
5. $\quad$ **If** $t > 0$ **then**
6. $\quad\quad \lambda_t = \text{Optimizer}(\nabla_{\lambda_{t-1}}\mathcal{L}_t, \lambda^*)$
7. $\quad \theta_{t+1} = \text{Optimizer}(\nabla_{\theta_t}\mathcal{L}_t, \lambda_t)$
8. $\quad a_{t+1} = \text{Optimizer}(\nabla_{a_t}\mathcal{L}_t, \lambda_t)$
9. 返回: θ_{T+1}, a_{T+1}, $\mathcal{B}_{T+1}$, λ_T.

算法 1 的动机在于: 对每个时刻 t, 网络自适应学习并获得用以更新参数的学习率. 对于算法初始化, 首先定义网络参数 θ_0, BN 统计量 $\mathcal{B}_0$, 网络参数的初始学习率 λ_0, 以及元学习率 λ^*. 对每个网络参数均设定对应的学习率, 因此, λ_0 和 λ^* 与网络参数 θ 维度相同. 第 3 行表示在 t 时刻, 给定当前网络参数 θ_t, 输入图像对 I_t, 则预测的深度图为 $D_t = (d_r, d_l) = \Phi((\theta_t, a_t, \mathcal{B}_t), I_t)$. 此处 d_l 和 d_r 分别定义左图像与右图像对应的深度图. 在前馈过程中, 算法使用第 2.1 节中描述的特征分布对齐算法, 动量为 a_t. 统计量 $\mathcal{B}_{t+1}$ 储存至下一时刻使用. 第 4 行表示预测的深度图由损失函数 $\mathcal{L}(D_t, I_t)$ 进行评估. 注意到学习率 λ_{t-1} 为 t 时刻损失 $\mathcal{L}_t$ 的参数, 因此, 在 t 时刻前馈完成后, 通过损失 $\mathcal{L}_t$ 求解针对 λ_{t-1} 的梯度, 并以元学习率 λ^* 进行梯度下降即可获得针对 t 时刻用于更新参数的学习率 λ_t. 以上过程在第 6 行表示. 由此, 算法基于上一时刻的网络参数学习率 λ_{t-1} 和当前时刻损失 $\mathcal{L}_t$, 得到了更适合于当前时刻场景的 λ_t 用于参数更新. 最后, 在第 7 行和第 8 行, 算法基于 λ_t 对网络参数 θ_t 和动量 a_t 进行梯度下降更新. 通过该算法, 每一时刻当新的场景输入模型后, 优化器以先前时刻学习率为经验, 以当前时刻的自监督损失为引导获得适用于当前时刻的参数学习率, 使网络中每个参数均得到合理更新, 帮助模型快速适应于当前场景. 在 OMLA 过程结束后, 模型获得 θ_{T+1}, a_{T+1}, $\mathcal{B}_{T+1}$ 和 λ_T.

2.3　基于元学习的模型预训练方法

本部分介绍一种新的基于元学习的预训练方法, 使模型在源域数据的预训练过程中获得针对视频流的快速收敛能力. 算法动机在于模拟在线学习过程, 通过在未来帧上计算损失, 寻找适用于 OMLA 算法在线更新的初始参数. 为实现这一目的, 算法首先利用 OMLA 方法使模型在一段连续视频流上进行在线适应过程, 然后根据模型在该视频流未来帧上的表现, 更新模型的初始参数. 形式上, 假设源域数据为 $\mathcal{S} = \left\{\{I_t^n\}_{t=0}^{T}\right\}_{n=1}^{N}$, 由 N 个长度为 T 的视频序列构成. 为简洁起见, 此处对每个视频序列均使用相同长度, 但在实际使用中源域视频序列的长度可以是任意的. 算法的目标是寻找网络模型权重 θ, 使得模型在视频流上进行 t' 步 OMLA 在线学习过程后, 在未来帧取得较低的损失函数, 即 $\sum_{t=t'+1}^{T}\mathcal{L}(\Phi(D, \theta_t, a_t), I_t)$ 的值较低. 重要的是, θ 需要在使用特征分布对齐方法的同时, 使网络快速收敛于当前视频流. 算法的具体步骤在算法 2 中展示.

算法 2. 基于元学习的模型预训练方法

1. 过程: $\{\{I_t^k\}_{t=0}^T\}_{k=1}^K$, θ, a, $\mathcal{B}$, λ, λ_{meta}
2. $grad_\theta = grad_\lambda = grad_a = 0$
3. 对 $k = 1$ to K 执行:
4. θ^k, a^k, $\mathcal{B}^k$, $\lambda^k = \text{OMLA}(\{I_t^k\}_{t=0}^{t'}, \theta, a, \mathcal{B}, \lambda, \lambda_{\text{meta}})$
5. 对 $t = t'+1$ to T 执行:
6. $D_t = \Phi\left((\theta^k, a^k, \mathcal{B}^k), I_t^k\right)$
7. $\mathcal{L}_t = \mathcal{L}(D_t, I_t)$
8. $grad_\theta += \nabla_\theta \mathcal{L}_t$
9. $grad_a += \nabla_a \mathcal{L}_t$
10. $grad_\lambda += \nabla_\lambda \mathcal{L}_t$
11. $\lambda = \text{Optimizer}(grad_\lambda, \lambda_{\text{meta}})$
12. $\theta = \text{Optimizer}(grad_\theta, \lambda_{\text{meta}})$
13. $a = \text{Optimizer}(grad_a, \lambda_{\text{meta}})$
14. 返回: (θ, a, λ).

算法 2 描述了一个训练步骤中的过程, 输入 $\{\{I_t^k\}_{t=0}^T\}_{k=1}^K \subset \mathcal{S}$ 由 K 个视频组成. 此处的 K 个视频是随机从 $\mathcal{S}$ 中选择的, 且互不相同. 训练步骤的初始状态包含网络初始参数 θ, 学习率 λ, BN 层动量 a 和统计量 $\mathcal{B}$. 元学习率 λ_{meta} 用于更新 θ, λ 和 a. 第 4 行表示对每个批次中的每个视频序列, 首先在其前 $0, \cdots, t'$ 帧上, 以算法 1 进行在线适应过程, 使模型适应于当前视频. 对第 k 个视频, 经过该过程后可以得到对应的网络参数 θ^k, 动量 a^k, BN 层统计量 $\mathcal{B}^k$ 和学习率 λ^k. 此处为简洁起见, 仅以视频标号 k 表示参数, 省略时间下标. 在实际场景中, 系统总是希望模型在之后的未来帧上取得较好的结果. 因此, 在第 5 ~ 7 行, 在视频剩余的未来帧 $(I_{t'+1}^k, \cdots, I_T^k)$ 上评估深度预测效果. 由于 θ^k, a^k, $\mathcal{B}^k$, λ^k 均由初始参数 θ, a, $\mathcal{B}$, λ 获得, 因此在第 8 ~ 10 行可计算未来帧损失 $\mathcal{L}_t$ 针对初始参数的梯度, 并逐一累加. 上述过程进行 K 个视频序列后, 在第 11 ~ 13 行, 算法通过 λ_{meta} 对原始参数 θ, λ 和 a 进行梯度下降并更新. 这种做法的目的在于寻找合适的初始化参数, 使模型在短暂的在线适应后, 在未来帧上表现较好, 即 $\mathcal{L}_t$ 值较低. 算法 2 描述的过程反复进行多次后, 获得的 θ, λ 和 a 可用作模型初始化参数, 在目标视频上进行在线适应以获得更好的深度估计效果.

2.4 深度估计损失函数

本节介绍在算法 1 和算法 2 中使用的自监督深度估计损失函数. 与文献 [9] 相同的是, 模型以双目图像 I_l, I_r 为输入并获得对应的视差图 d_l, d_r. 使用扭转操作 f_w, 通过视差图 d_l 从右图像重构左图像:

$$\hat{I}_l = f_w(I_r, d_l) \tag{4}$$

对称地, 可用相同的方法从左图像获得右图像的重构结果. 计算两个重构图像与原图像的误差可以反映深度预测的效果. 最终的损失函数由 $\mathcal{L}_1$ 重构误差和基于结构相似性 (Structural similarity index, SSIM) 的自重构误差 $\mathcal{L}_{\text{SSIM}}$ (参见文献 [9]) 组成:

$$\begin{aligned}\mathcal{L} = {} & (1-\alpha)||\hat{I}_l - I_l||_1 + \alpha\mathcal{L}_{\text{SSIM}}(\hat{I}_l, I_l) + \\ & (1-\alpha)||\hat{I}_r - I_r||_1 + \alpha\mathcal{L}_{\text{SSIM}}(\hat{I}_r, I_r)\end{aligned} \tag{5}$$

使用上述损失函数进行训练, 模型能够无监督地获得深度估计能力, 进行在线适应过程时也无需深度真值.

3 相关实验

3.1 评估方法和标准

1) 在线适应效果的评估方法. 在线学习或适应过程中, 视频帧被序列化地输入模型. 对每个视频帧, 模型均进行深度估计和参数更新. 需要强调的是, 在实验中为了合理地评估模型效果, 在模型获得预测结果后立即进行评估, 然后进行参数更新. 在整个视频输入完毕后, 计算针对该视频的平均效果以评估模型在该视频上的整体表现, 同时计算最后 20% 视频帧的平均效果以评估模型在最后阶段是否已经较好地收敛.

2) 评估准则. 本文采用与文献 [1, 9, 40] 相同的评估准则. 设 P 为测试视频中具有深度真值的像素点总数, $\hat{d}_i$, d_i 分别为在像素 i 处的预测结果和深度真值, 则评估准则如下:

a) 平均相对误差 (Mean relative error, Abs Rel): $(\frac{1}{P}\sum_{i=1}^{P}(\| \hat{d}_i - d_i \|))/d_i$;

b) 平方相对误差 (Squared relative error, Sq Rel): $(\frac{1}{P}\sum_{i=1}^{P}(\| \hat{d}_i - d_i \|^2))/d_i$;

c) 均方误差 (Root mean squared error, RMSE): $\sqrt{\frac{1}{P}\sum_{i=1}^{P}(\hat{d}_i - d_i)^2}$;

d) 对数均方误差 (Mean squared logarithmic error, $\text{RMSE}_{\log}$): $\sqrt{\frac{1}{P}\sum_{i=1}^{P} \| \log \hat{d}_i - \log d_i \|^2}$;

e) 以 τ 为阈值的精度, 即满足条件 $\delta = \max(d_i/\hat{d}_i, \hat{d}_i/d_i) < \alpha^\tau$ 的预测结果 $\hat{d}_i$ 所占比例. 与文献 [1] 相同, 其中, $\alpha = 1.25$, $\tau \in [1, 2, 3]$.

3.2 数据库与部署细节

本文实验以包含视频序列的自动驾驶合成数据库为源域数据库, 在该数据库上进行预训练. 以现实驾驶场景为目标域数据库, 在该数据库上进行在

线适应和评估. 所使用的数据库如下:

1) Synthia[20] 为一个城市驾驶场景的合成数据库. 其包含 5 个不同场景和季节的双目视频流. 本文使用春季场景中模拟汽车以前置摄像头记录的视频流, 共包含 4000 对双目图像, 并在该数据上进行预训练过程.

2) Scene Flow Driving[41] 为一个驾驶场景的合成视频数据库. 其包含不同的相机设定、车速和行驶方向. 本文选择汽车前进场景中所有的视频数据, 共 2000 对双目图像.

3) KITTI[14] 为现实行车场景数据库. 本文选择文献 [1] 提供的子集 (简称 Eigen 子集) 为目标域数据. 该子集包含 32 个不同的训练场景以及 28 个不同的测试场景. 实验中以 28 个测试视频进行在线适应和评估.

本文以 Pytorch[42] 平台部署方法, 以一块 Nividia P40 GPU 进行训练和测试. 在实验中, 每个网络模型均在卷积层后包含 BN 层[43], 以进行特征分布对齐. 在合成数据集上的预训练过程中, 首先以 10^{-4} 为学习率进行 100 个周期的训练, 然后利用算法 2 进行 10 个周期的元预训练过程. 此处设定 $\lambda = 10^{-4}$, $\lambda_{\text{meta}} = 10^{-5}$, $K = 8$, 视频长度 $T = 8$, $t' = 5$. 在目标视频的在线适应过程中, 使用 OMLA 算法 (即算法 1), 元学习率设置为 $\lambda^* = 10^{-7}$, 特征分布对齐方法仅在编码网络的 BN 层中进行. 所有的训练和测试过程使用 Adam 优化器[44].

3.3　实验结果与分析

3.3.1　算法消融实验

为了验证本文算法的有效性, 本节选择在常用的双目深度估计框架 (在文献 [9] 中提出) 上进行算法消融实验. 值得一提的是, 尽管该框架最初是作为单目深度估计方法而提出的, 但随后在文献 [9–10] 中成功用于双目深度估计. 除特别说明, 所有模型均在 Synthia 数据集上进行预训练, 并且在 Eigen 子集的测试集视频上进行在线适应和评估. 每个视频序列被单独评估效果, 即均从 Synthia 数据集预训练得到的初始网络参数开始在线适应过程. 最终的效果得分为所有视频帧的平均得分.

相关的实验结果在表 1 中展示. 其中, 基准方法表示仅使用一般梯度下降法的在线适应模型, 在线特征分布对齐表示仅使用第 2.1 节中方法, 且采用基准方法进行在线适应的模型. 标准预训练方法表示不使用第 2.3 节中元预训练方法, 仅采用一般训练方法的模型. 首先可以观察到, 不使用预训练过程的模型难以直接在目标域视频流上取得令人满意的效果, 而使用基准方法对效果的提升也相当有限. 这证明了仅在源域训练时, 或者数据间存在域转移时会导致算法效果的显著下降. 当模型使用在线特征分布对齐方法后, 其效果有明显提升, 这表明其能够较好地处理在线适应过程中的域转移问题. 使用 OMLA 算法使结果进一步提升. 同时, 使用元预训练方法的模型, 其在线适应效果均好于使用标准预训练方法的模型. 这些结果证明了本文元学习算法的有效性.

3.3.2　不同网络模型和数据库的实验

为了进一步验证本文方法的有效性, 本节首先在 3 种网络框架下进行对比实验, 分别为: DispNet[41], MADNet[15] 和 ResNet[9]. 实验主要比较了基准方法和 OMLA + 元预训练算法的效果, 其中基准方法在源域上进行标准预训练过程. 同时, 为了验证本文方法在不同数据库上的普适性, 本节分别在 Synthia 和 Scene Flow Driving 数据库上进行预训练以比较效果. 相关实验结果在表 2 中展示. 可以看到, 在每个网络模型中, OMLA 算法均取得了明显的提升. 这证明了本文方法在小型网络 (如 DispNet 和 MADNet) 中仍能取得良好效果. 在两个数据库上, OMLA 算法均带来了提升, 证明了其在数据库间的

表 1　KITTI Eigen 测试集的算法消融实验 (仅评估 50 m 之内的深度估计效果)

Table 1　Ablation study on KITTI Eigen test set (the results are evaluated within 50 m)

方法	预训练方式	平均得分				最后 20% 帧的平均得分			
		RMSE	Abs Rel	Sq Rel	$\text{RMSE}_{\log}$	RMSE	Abs Rel	Sq Rel	$\text{RMSE}_{\log}$
基准方法	无	12.2012	0.4357	5.5672	1.3598	12.2874	0.4452	5.5213	1.3426
基准方法	标准预训练方法	9.0518	0.2499	3.2901	0.9503	9.0309	0.2512	3.3104	0.9495
仅在线特征分布对齐	标准预训练方法	3.6135	0.1250	0.6972	0.2041	3.5857	0.1031	0.6887	0.1910
OMLA 算法	标准预训练方法	3.5027	0.0923	0.6611	0.1896	3.3986	0.0882	0.6579	0.1735
基准方法	元预训练方法	8.8230	0.2305	3.0578	0.9324	8.7061	0.2273	2.9804	0.9065
仅在线特征分布对齐	元预训练方法	3.5043	0.0950	0.6627	0.1992	3.4831	0.0896	0.6545	0.1921
OMLA 算法	元预训练方法	**3.4051**	**0.0864**	**0.6256**	**0.1852**	**3.3803**	**0.0798**	**0.6176**	**0.1801**

表 2 不同网络模型和数据库上的结果对比

Table 2 Comparison on different network architectures and datasets

网络模型	方法	预训练于 Synthia[20]				预训练于 Scene Flow Driving[41]				帧速率 (帧/s)
		RMSE	Abs Rel	Sq Rel	$\text{RMSE}_{\log}$	RMSE	Abs Rel	Sq Rel	$\text{RMSE}_{\log}$	
ResNet[9]	基准方法	9.0518	0.2499	3.2901	0.8577	9.0893	0.2602	3.3896	0.8901	**5.06**
	OMLA + 元预训练	**3.4051**	**0.0864**	**0.6256**	**0.1852**	**4.0573**	**0.1231**	**1.1532**	**0.1985**	3.40
MADNet[15]	基准方法	8.8650	0.2684	3.1503	0.8233	8.9823	0.2790	3.3021	0.8350	**12.05**
	OMLA + 元预训练	**4.0236**	**0.1756**	**1.1825**	**0.2501**	**4.2179**	**0.1883**	**1.2761**	**0.2523**	9.56
DispNet[41]	基准方法	9.0222	0.2710	4.3281	0.9452	9.1587	0.2805	4.3590	0.9528	**5.42**
	OMLA + 元预训练	**4.5201**	**0.2396**	**1.3104**	**0.2503**	**4.6314**	**0.2457**	**1.3541**	**0.2516**	4.00

普适性. 表 2 中同时比较了算法的运行时间, 以帧速率 (帧/s) 为指标. 可以看到, OMLA 算法由于需要额外的梯度回传过程, 因此相比于基准方法, 其帧速率降低了大约 20%. 然而, 考虑到取得的效果提升, 该运行时间是可以接受的.

3.3.3 与理想模型和当前最佳方法的比较

本部分通过实验比较本文提出的算法与理想模型 (即利用目标域数据训练的模型) 的在线适应效果. 实验中的网络模型均为以 ResNet 构建的模型[9]. 为训练理想模型, 首先将模型在 KITTI Eigen 训练子集上进行预训练. 这样一来, 当模型用于目标域数据, 即 KITTI Eigen 测试子集时, 不会受到域转移问题的影响. 对于理想模型的分析, 采用了两种做法: 1) 在目标域上不进行在线适应和模型更新; 2) 在目标域以基准方法进行在线适应和模型更新. 同时, 实验比较了 L2A 算法[35], 该算法为当前双目深度估计领域最佳的在线适应方法. 相关结果在表 3中展示, 除 OMLA+元预训练方法外, 其余模型均采用标准预训练方法. 可以看到, 当模型在目标域进行预训练后, 基准方法无法增强在线适应的效果. 进一步地, 相比于在目标域预训练的理想模型, 尽管存在域转移问题, OMLA 算法获得了有竞争力的在线适应效果. 相比于 L2A 算法, 尽管其使用基准方法进行在线适应, 帧速率略高于 OMLA 算法, 但 OMLA 算法的准确度较之获得了明显提升, 这证明了处理在线适应过程中域转移问题的重要性, 也进一步验证了本文方法的效果. 以上结果表明, 尽管模型在训练过程中从未接触真实世界数据, 其仍然可以利用本文的在线适应方法获得针对目标场景的快速收敛能力, 并逐渐预测精准的场景深度图.

为了进一步分析本文方法, 图 3 展示了不同模型的深度估计视觉效果. 为了评估不同模型的在线适应效果随时间的变化, 此处展示了视频初始、中段和末段时刻的深度估计效果. 可以看到, 理想模型在 3 个时期的效果均是良好的, 这是由于理想模型在目标域进行训练, 因此不存在域转移问题, 模型的在线适应效果可以得到保证. L2A 算法也取得了不错的效果, 但其问题也较为明显, 例如在初始阶段无法预测出道路树木、交通标志杆的深度, 在

表 3 与理想模型和当前最优方法的比较 (仅比较实际深度值小于 50 m 的像素点)

Table 3 Comparison with ideal models and state-of-the-art method (Results are only evaluated within 50 m)

网络模型	在线适应算法	预训练域	RMSE	Abs Rel	Sq Rel	$\text{RMSE}_{\log}$	$\alpha > 1.25$	$\alpha > 1.25^2$	$\alpha > 1.25^3$
ResNet[9]	无	目标域	3.6975	0.0983	1.1720	0.1923	0.9166	0.9580	0.9778
	基准方法	目标域	3.4359	**0.0850**	0.6547	0.1856	**0.9203**	0.9612	0.9886
	L2A[35]	源域	3.5030	0.0913	0.6522	**0.1840**	0.9170	0.9611	0.9882
	OMLA+元预训练	源域	**3.4051**	0.0864	**0.6256**	0.1852	0.9170	**0.9623**	**0.9901**
MADNet[15]	无	目标域	**3.8965**	0.1793	1.2369	**0.2457**	0.9147	0.9601	0.9790
	基准方法	目标域	3.9023	0.1760	1.1902	0.2469	**0.9233**	0.9652	0.9813
	L2A[35]	源域	4.1506	0.1788	1.1935	0.2533	0.9131	0.9443	0.9786
	OMLA+元预训练	源域	4.0236	**0.1756**	**1.1825**	0.2501	0.9022	**0.9658**	**0.9842**
DispNet[41]	无	目标域	4.5210	0.2433	**1.2801**	**0.2490**	0.9126	0.9472	**0.9730**
	基准方法	目标域	4.5327	**0.2368**	1.2853	0.2506	**0.9178**	**0.9600**	0.9725
	L2A[35]	源域	4.6217	0.2410	1.2902	0.2593	0.9062	0.9513	0.9688
	OMLA+元预训练	源域	**4.5201**	0.2396	1.3104	0.2503	0.9085	0.9460	0.9613

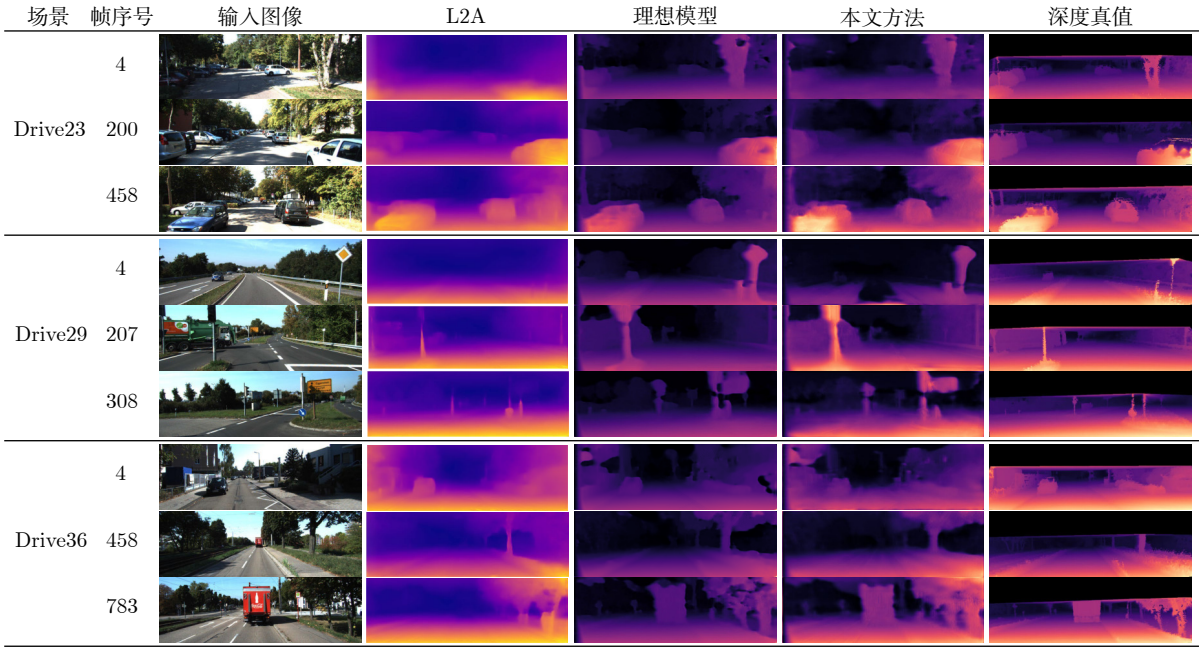

图 3 在 KITTI Eigen 测试集中 3 个不同视频序列上的效果 (为了展示模型的在线适应效果随时间的变化, 此处展示了视频初始, 中段和末段时刻的深度估计效果)

Fig. 3 Performance on three different videos of KITTI Eigen test (We illustrated predictions of initial, medium, and last frames)

中段和后段依然存在大目标 (如大型标志牌、大货车) 的深度不准确、丢失等问题. 相比之下, 本文方法在视频初期预测效果有限, 但在中段和末段帧上的表现较好, 接近理想模型和深度真值. 这是由于在视频开始阶段, 模型仍然受到场景差异的影响而难以获得准确的预测结果. 当进行一段时间的在线适应后, 本文方法使模型较好地收敛至当前视频场景, 因而预测效果获得了明显提升. 以上的视觉效果展示较好地说明了本文方法对模型在线适应效果的提升.

4 结束语

本文研究了深度估计模型的在线适应问题, 并提出了一种新的在线元学习适应算法 OMLA. 该算法适用于需要深度估计网络模型快速收敛和适应于目标视频流的在线序列化学习场景. 在公开数据集上的大量实验表明, 相对于一般的梯度下降算法, 本文方法较好地处理了域转移问题, 并通过元学习方法充分利用每一时刻的学习反馈, 有助于提升深度估计模型的在线适应效果. 与当前最好的方法相比, 本文方法取得了明显提升; 与理想模型相比, 本文方法获得了接近的并有竞争力的效果.

References

1 Eigen D, Fergus R. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). Santiago, Chile: IEEE, 2015. 2650–2658

2 Laina I, Rupprecht C, Belagiannis V, Tombari F, Navab N. Deeper depth prediction with fully convolutional residual networks. In: Proceedings of the 4th International Conference on 3D Vision (3DV). Stanford, USA: IEEE, 2016. 239–248

3 Fu H, Gong M M, Wang C H, Batmanghelich K, Tao D C. Deep ordinal regression network for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA: IEEE, 2018. 2002–2011

4 Xu D, Wang W, Tang H, Liu H, Sebe N, Ricci E. Structured attention guided convolutional neural fields for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA: IEEE, 2018. 3917–3925

5 Zhang Z Y, Cui Z, Xu C Y, Jie Z Q, Li X, Yang J. Joint task-recursive learning for semantic segmentation and depth estimation. In: Proceedings of the 15th European Conference on Computer Vision. Munich, Germany: Springer, 2018. 235–251

6 Wofk D, Ma F C, Yang T J, Karaman S, Sze V. FastDepth: Fast monocular depth estimation on embedded systems. In: Proceedings of the International Conference on Robotics and Automation (ICRA). Montreal, Canada: IEEE, 2019. 6101–6108

7 Ma F C, Karaman S. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Brisbane, Australia: IEEE, 2018. 1–8

8 Garg R, Kumar B G V, Carneiro G, Reid I. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In: Proceedings of the 14th European Conference on Computer Vision. Amsterdam, the Netherlands: Springer, 2016. 740–756

9 Godard C, Aodha O M, Brostow G J. Unsupervised monocular depth estimation with left-right consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recogni-

tion (CVPR). Honolulu, USA: IEEE, 2017. 270−279

10 Pilzer A, Xu D, Puscas M, Ricci E, Sebe N. Unsupervised adversarial depth estimation using cycled generative networks. In: Proceedings of the International Conference on 3D Vision (3DV). Verona, Italy: IEEE, 2018. 587−595

11 Pillai S, Ambruş R, Gaidon A. SuperDepth: Self-supervised, super-resolved monocular depth estimation. In: Proceedings of the International Conference on Robotics and Automation (ICRA). Montreal, Canada: IEEE, 2019. 9250−9256

12 Mancini M, Karaoguz H, Ricci E, Jensfelt P, Caputo B. Kitting in the wild through online domain adaptation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Madrid, Spain: IEEE, 2018. 1103−1109

13 Carlucci F M, Porzi L, Caputo B, Ricci E, Bulò S R. AutoDIAL: Automatic domain alignment layers. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). Venice, Italy: IEEE, 2017. 5077−5085

14 Menze M, Geiger A. Object scene flow for autonomous vehicles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, USA: IEEE, 2015. 3061−3070

15 Tonioni A, Tosi F, Poggi M, Mattoccia S, Di Stefano L. Real-time self-adaptive deep stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, USA: IEEE, 2019. 195−204

16 Liu F Y, Shen C H, Lin G S, Reid I. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016, **38**(10): 2024−2039

17 Yang N, Wang R, Stückler J, Cremers D. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In: Proceedings of the 15th European Conference on Computer Vision. Munich, Germany: Springer, 2018. 835−852

18 Silberman N, Hoiem D, Kohli P, Fergus R. Indoor segmentation and support inference from RGBD images. In: Proceedings of the 12th European Conference on Computer Vision. Florence, Italy: Springer, 2012. 746−760

19 Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Providence, USA: IEEE, 2012. 3354−3361

20 Ros G, Sellart L, Materzynska J, Vazquez D, Lopez A M. The SYNTHIA dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, USA: IEEE, 2016. 3234−3243

21 Zhou T H, Brown M, Snavely N, Lowe D G. Unsupervised learning of depth and ego-motion from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, USA: IEEE, 2017. 1851−1858

22 Kundu J N, Uppala P K, Pahuja A, Babu R V. AdaDepth: Unsupervised content congruent adaptation for depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA: IEEE, 2018. 2656−2665

23 Csurka G. Domain adaptation for visual applications: A comprehensive survey. arXiv preprint arXiv: 1702.05374, 2017.

24 Long M S, Zhu H, Wang J M, Jordan M I. Deep transfer learning with joint adaptation networks. In: Proceedings of the 34th International Conference on Machine Learning. Sydney, Australia: JMLR.org, 2017. 2208−2217

25 Venkateswara H, Eusebio J, Chakraborty S, Panchanathan S. Deep hashing network for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, USA: IEEE, 2017. 5018−5027

26 Bousmalis K, Irpan A, Wohlhart P, Bai Y F, Kelcey M, Kalakrishnan M, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Brisbane, Australia: IEEE, 2018. 4243−4250

27 Sankaranarayanan S, Balaji Y, Castillo C D, Chellappa R. Generate to adapt: Aligning domains using generative adversarial networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA: IEEE, 2018. 8503−8512

28 Wulfmeier M, Bewley A, Posner I. Incremental adversarial domain adaptation for continually changing environments. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). Brisbane, Australia: IEEE, 2018. 1−9

29 Li Y H, Wang N Y, Shi J P, Liu J Y, Hou X D. Revisiting batch normalization for practical domain adaptation. In: Proceedings of the 5th International Conference on Learning Representations. Toulon, France: OpenReview.net, 2017.

30 Mancini M, Porzi L, Bulò S R, Caputo B, Ricci E. Boosting domain adaptation by discovering latent domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA: IEEE, 2018. 3771−3780

31 Tobin J, Fong R, Ray A, Schneider J, Zaremba W, Abbeel P. Domain randomization for transferring deep neural networks from simulation to the real world. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Vancouver, Canada: IEEE, 2017. 23−30

32 Tonioni A, Poggi M, Mattoccia S, Di Stefano L. Unsupervised adaptation for deep stereo. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). Venice, Italy: IEEE, 2017. 1605−1613

33 Pang J H, Sun W X, Yang C X, Ren J, Xiao R C, Zeng J, et al. Zoom and learn: Generalizing deep stereo matching to novel domains. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA: IEEE, 2018. 2070−2079

34 Zhao S S, Fu H, Gong M M, Tao D C. Geometry-aware symmetric domain adaptation for monocular depth estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, USA: IEEE, 2019. 9788−9798

35 Tonioni A, Rahnama O, Joy T, Di Stefano L, Ajanthan T, Torr P H S. Learning to adapt for stereo. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, USA: IEEE, 2019. 9661−9670

36 Vinyals O, Blundell C, Lillicrap T, Kavukcuoglu K, Wierstra D. Matching networks for one shot learning. In: Proceedings of the 30th International Conference on Neural Information Processing Systems. Barcelona, Spain: Curran Associates Inc., 2016. 3637−3645

37 Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the 34th International Conference on Machine Learning. Sydney, Australia: JMLR.org, 2017. 1126−1135

38 Guo Y H, Shi H H, Kumar A, Grauman K, Rosing T, Feris R. SpotTune: Transfer learning through adaptive fine-tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, USA: IEEE, 2019. 4805−4814

39 Park E, Berg A C. Meta-tracker: Fast and robust online adaptation for visual object trackers. In: Proceedings of the 15th European Conference on Computer Vision. Munich, Germany: Springer, 2018. 587−604

40 Wang P, Shen X H, Lin Z, Cohen S, Price B, Yuille A. Towards unified depth and semantic prediction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, USA: IEEE, 2015. 2800−2809

41 Mayer N, Ilg E, Häusser P, Fischer P, Cremers D, Dosovitskiy A, et al. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, USA: IEEE, 2016. 4040−4048

42 Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, et al. Automatic differentiation in PyTorch. In: Proceedings of the 31st Conference on Neural Information Processing Systems. Long Beach, USA: 2017.

43 Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning. Lille, France: JMLR.org, 2015. 448−456

44 Kingma D P, Ba J. Adam: A method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations. San Diego, USA: 2015.

张振宇 南京理工大学计算机科学与工程学院 PCA 实验室博士研究生. 2015 年获得南京理工大学理学院信息与计算科学系学士学位. 主要研究方向为基于视觉的深度估计方法, 深度学习算法.
E-mail: zhangjesse@njust.edu.cn
(**ZHANG Zhen-Yu** Ph.D. candidate at PCA Laboratory, School of Computer Science and Engineering, Nanjing University of Science and Technology. He received his bachelor degree in 2015. His research interest covers computer vision and deep learning, specially on depth estimation.)

杨 健 南京理工大学计算机科学与工程学院教授, 长江学者, IAPR Fellow. 主要研究方向为矩阵回归, 自动驾驶和机器人场景的视觉感知. 本文通信作者.
E-mail: csjyang@njust.edu.cn
(**YANG Jian** Professor at the School of Computer Science and Engineering, Nanjing University of Science and Technology. He is also a Changjiang Scholar and IAPR Fellow. His research interest covers matrix regression, visual perception in autonomous driving and robotics. Corresponding author of this paper.)