基于三维卷积神经网络的立体匹配算法

王玉锋[1,2], 王宏伟[2,3**], 于光[2], 杨明权[2], 袁昱纬[4], 全吉成[1,2*]

[1]海军航空大学, 山东 烟台 264001;
[2]空军航空大学, 吉林 长春 130022;
[3]信息工程大学, 河南 郑州 450001;
[4]91977 部队, 北京 102200

摘要 对于基于深度学习的立体匹配而言,模型的网络结构对算法精度的影响很大,而算法运行效率也是实际应用中需要考虑的重要因素。提出一种在视差维度上使用稀疏损失体进行立体匹配的方法。采用宽步长平移右视角特征图构建稀疏的三维损失体,使三维卷积模块所需的显存和计算资源均降低数倍。采用多类别输出的方式对匹配损失在视差维度上进行非线性上采样,并结合两种损失函数训练模型,在保证运行效率的同时提高算法精度。在 KITTI 测试集上,与基准算法相比,所提算法不仅提高了精度,而且运行时间缩短了约 40%。

关键词 机器视觉; 立体匹配; 深度学习; 双目视觉; 卷积神经网络

中图分类号 TP391.4 **文献标识码** A **doi**: 10.3788/AOS201939.1115001

Stereo Matching Algorithm Based on Three-Dimensional Convolutional Neural Network

Wang Yufeng[1,2], Wang Hongwei[2,3**], Yu Guang[2], Yang Mingquan[2], Yuan Yuwei[4], Quan Jicheng[1,2*]

[1]*Naval Aviation University, Yantai, Shandong* 264001, *China*;
[2]*Aviation University of Air Force, Changchun, Jilin* 130022, *China*;
[3]*Information Engineering University, Zhengzhou, Henan* 450001, *China*;
[4]*The* 91977 *Troops, Beijing* 102200, *China*

Abstract For a stereo matching method based on deep learning, network architecture is critical to ensuring the algorithm's accuracy; efficiency is also an important factor to consider in practical applications. A stereo matching method with a sparse cost volume in the disparity dimension is proposed herein. The three-dimensional sparse cost volume is created by shifting right-view features with a large step to substantially reduce the memory and computational resources in a three-dimensional convolution module. The matching cost is nonlinearly sampled via multiclass output in the disparity dimension, and the model is trained by merging two types of loss functions, such that the proposed method's accuracy is improved without any notable reduction in efficiency. The proposed algorithm reduces running time by about 40% while improving accuracy compared with the benchmark algorithm on the KITTI test dataset.

Key words machine vision; stereo matching; deep learning; binocular vision; convolutional neural network

OCIS codes 150.6910; 150.5670; 150.1135

1 引 言

立体匹配是计算机视觉中的一个基础性研究,广泛应用于三维重构、无人驾驶及机器人导航等多种领域。传统的立体匹配算法多围绕损失计算和视差优化进行研究:一方面,设计良好的度量函数来计算匹配损失[1-2];另一方面,使用局部或全局的方法为每个像素分配视差值[3-4]。这些算法均采用人工设计的浅函数,对于病态区域(如纹理少的区域等)往往不能得到正确的结果。

收稿日期: 2019-05-05; **修回日期**: 2019-06-21; **录用日期**: 2019-07-08

基金项目: 国家杰出青年科学基金(61301233)

* **E-mail**: jicheng_quan@126.com; ** **E-mail**: alex19820911@126.com

近年来，深度学习表现出了强大的图像理解能力，在目标分类、目标检测和语义分割等任务中具有优异的性能[5-7]，基于深度学习的立体匹配算法也越来越受关注[8-14]。卷积神经网络(CNN)可从图像中提取稳健特征，很适于学习图像块之间的相似度[8-10]。在匹配性模糊的区域，为进一步提高模型的全局优化能力，端到端的立体匹配方法[11-12]将视差预测的全过程整合到 CNN 模型中。然而，这种算法多采用一种沿视差方向的一维相关算法，损失了在视差维度的特征。在立体匹配中引入三维卷积神经网络(3DCNN)[13-14]，使模型可以在 3 个维度上去理解全局语义信息，从而能更好地理解场景的上下文信息。

在立体匹配算法中引入 3DCNN，对匹配过程建模效果很好，但也使显存开销和计算量增加了数十倍。这些资源负载主要来自 3DCNN，因此，所提算法主要从降低这些负载入手，主要贡献包含 3 个方面：1)构建视差维度上稀疏损失体作为 3DCNN 的输入，降低显存开销和计算量；2)在单个平移步长内增加 3DCNN 的输出类别数，在视差维度上对匹配损失进行更精细的采样；3)在最大概率邻域内进行视差回归，并结合亚像素的交叉熵损失和平滑的 L1 损失进行训练，使模型不仅能对视差图进行更准确的亚像素估计，而且在直接扩展视差范围时对精度影响较小。

2　相关工作

典型的立体匹配算法通常包含 4 个步骤[15]：匹配损失计算、损失聚合、初始视差计算和视差细化。大量公开的提供高质量视差真值的立体匹配数据集[11,16-18]，不仅为各种立体匹配算法提供了定量的对比，而且为深度学习以各种方式改进立体匹配算法提供了可能。最初，CNN 被用于计算图像块之间的相似度，Žbontar 等[9-10]训练了一个孪生网络来提取稳健的图像特征和计算匹配损失，与传统方法相比其性能取得了较大的提升。Luo 等[19]将视差预测转化为多标签分类任务，使用点积来计算相似度打分，显著提高了算法速度。

端到端的学习在算法的整体优化上往往能获得更好的性能，Mayer 等[11]提出一种“编码-解码”网络结构，并创建了一个大型的合成数据集来进行视差的端到端学习。以此视差预测网络为基础，Pang 等[12]通过级联另一个网络进行视差微调以提高精度。Liang 等[20]将 CNN 与贝叶斯推理相结合，通过学习先验和后验的不变特征来进行视差预测和微调。Jie 等[21]引入循环神经网络，通过不断对比左右视角来逐渐改善视差预测结果。

为更好地利用语义的上下文信息，Kendall 等[13]使用 3DCNN 学习进行匹配损失的计算，再以可差分的回归函数进行亚像素的视差预测，使模型可以从 3 个维度上理解全局语义信息。在此基础上，Yu 等[22]引入一个明确的损失聚合子模块进行匹配损失优化。Smolyanskiy 等[23]根据双目图像的几何关系构建半监督的损失函数，以稀疏的视差真值来提供密集的误差反馈信号。Chang 等[14]采用平均池化模块进行多尺度特征融合，提高了特征抽象能力，并以深层监督的方式学习匹配损失计算。这些算法增加了新的视差维度，显存开销和计算量增加了数十倍，且场景变化需要扩展视差范围时仍需对模型进行重新训练或微调。

虽然现有研究的重点是有监督的立体匹配算法，但无监督的立体匹配算法[24-25]也是非常值得关注的内容，这种算法在训练过程中不需要大范围高质量的视差数据，根据左右图像的几何约束关系就能学习如何预测视差，从而大大减少采集训练数据所需的工作量。

3　基于 3DCNN 的立体匹配算法

立体匹配是从双目图像中找到同名对应点，输出密集视差图的过程，可以将其转化为端到端的值回归任务，以双目图像为输入，直接输出预测的视差图。所提算法的网络结构主要由 4 个部分组成，分别为特征提取、空间金字塔特征融合、匹配损失计算和视差回归，如图 1 所示。基本流程如下：1)使用 CNN 分别对左右视角图像进行特征提取，并融合多尺度特征；2)连接左视角特征和平移的右视角特征，构建视差维度上稀疏的损失体，再使用 3DCNN 学习并根据几何上下文信息计算匹配损失；3)重采样损失体到原始图片尺寸，用 Softmax 函数将损失值转化为视差概率分布，并通过视差回归函数输出亚像素的预测视差。

3.1　网络结构的改进

连接左视角特征和平移的右视角特征构建损失体的过程如图 1 下方的虚线框所示，其中，矩形表示左右视角特征，矩形的叠加表示对特征进行连接。通常，平移右视角特征图的步长(记为 S)为 1，如图 1 下方虚线框的上半部分所示，在最大视差(记为 D)范围内构建损失体，其数据量变为特征图的 D

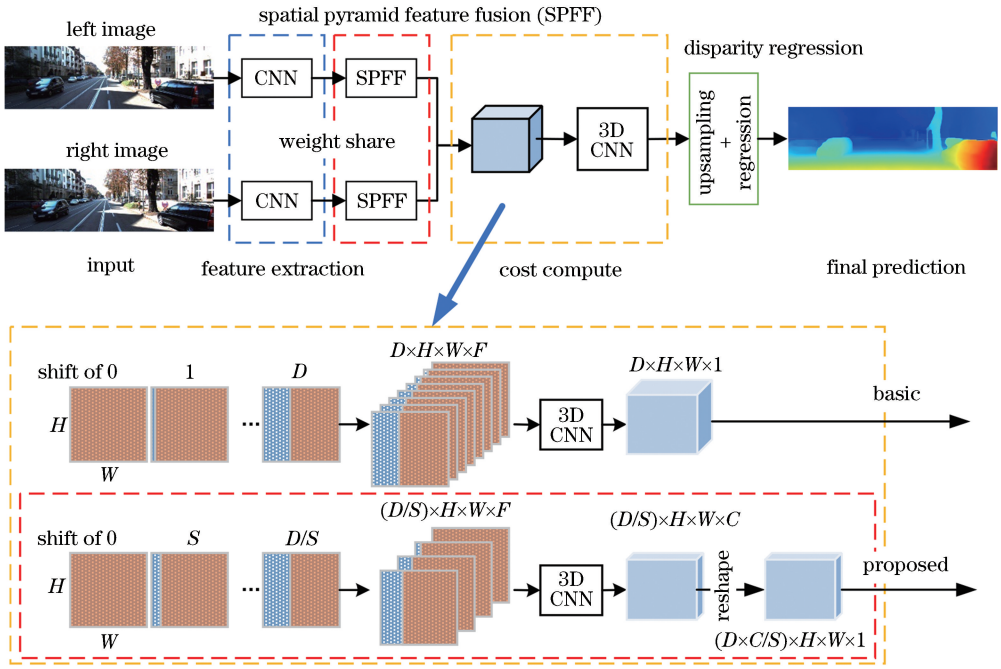

图 1　算法的网络结构

Fig. 1　Architecture overview of proposed method

倍，单个三维卷积层的计算量将是单个二维卷积层的 3D 倍（卷积核大小为 3）。

对图像特征在视差维度上进行小范围（S=1）的平移和堆叠，显然存在大量的信息冗余。采用相对较宽的平移步长（S>1），在视差维度稀疏的损失体上计算匹配损失，相对于平移步长为 1 的情况，三维卷积模块的显存开销和计算量均能降低为约 1/S。与 S=1 时相比，三维损失体在视差维度上只有原来的 1/S，而每个三维卷积层的特征维度不变，其输入输出也只有原来的 1/S，因此，所需要的显存和计算资源均为原来的 1/S。

卷积神经网络往往可以综合一定像素范围内的信息，同时也能对其进行分解。平移步长的增加使模型在视差维度上的细化能力变弱。为减弱这种影响，在每个平移步长内，对匹配损失进行多类别预测（其数目记为 C）。这相当于将匹配损失进行了非线性的上采样，使模型可以学习视差概率分布的细化函数，从而改善算法精度。由于只在匹配损失的输出层增加了权重，因此其对算法运行效率的影响较小。

不同的 S 值和 C 值决定了匹配损失在视差维度上的采样数，如图 2 所示。图 2(a)为基准方法的情况，在每个视差值都对应一个采样点；图 2(b)为宽步长平移的情况，每个平移步长产生一个采样点，虚线部分是由于宽步长平移而减少的采样点；图 2(c)为宽步长平移、多类别预测的情况，宽步长平移造成了采样点的减少，通过多类别预测进行采样点的补充，相当于在视差维度上对匹配损失进行了非线性上采样。

设置的 S 值越大，效率越高，而精度越低，因此需要权衡精度和效率，选择合适的 S 值。设置的 C 值越大，非线性上采样的因子就越大，潜在的模型细化能力就越强，但也会增加模型训练的难度，因此 C

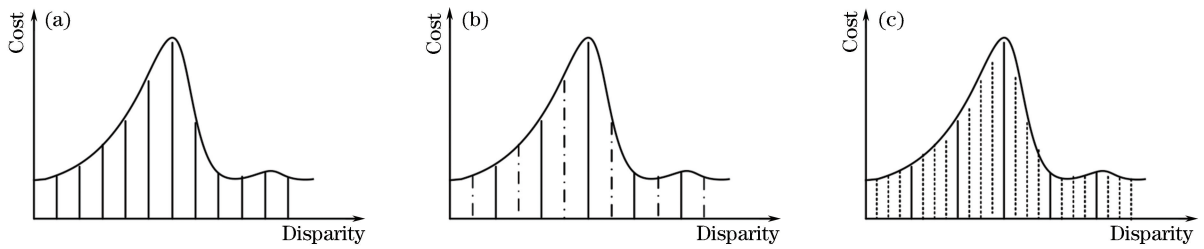

图 2　在视差维度上对损失进行采样的可视化描述。(a) S=1, C=1; (b) S=2, C=1; (c) S=2, C=4

Fig. 2　Graphical depiction of sampling cost in disparity dimension. (a) S=1, C=1; (b) S=2, C=1; (c) S=2, C=4

值的选择不仅会影响模型的收敛精度，而且会影响模型的收敛速度。

3.2　损失函数的改进

文献[14]中使用可差分的视差回归模块预测视差值，结合平滑的 L1(绝对值误差)损失[26]对模型进行训练。可差分的视差回归模块使用 Softmax 函数 $\sigma(\cdot)$，根据匹配损失 C 计算视差的概率分布，再以加权的方式对视差值进行亚像素估计，其表达式为

$$\hat{d}_{\mathrm{A}}=\sum_{n=0}^{N_{\mathrm{d}}}d_n\sigma(-C_n),\tag{1}$$

式中：$\hat{d}_{\mathrm{A}}$ 为视差值的亚像素估计；$d_n=D_{\max}n/N_{\mathrm{d}}$，$D_{\max}$为设置的最大视差，$N_{\mathrm{d}}$为视差维度的采样数，$n$ 为视差维度的索引。

当视差概率分布是单峰且对称时，(1) 式可以得到较好的亚像素估计，当视差分布存在多峰值时，预测值将会远离峰值。Kendall 等[13]阐述了 CNN 学习可以对输出值进行预尺度处理，并使其分布具有单峰性，但这种预尺度仅适合训练阶段使用的视差范围，当在测试阶段改变视差范围时，需要对模型参数进行重新训练或微调。若只在最大概率的视差值邻域内进行加权平均，可以有效解决上述问题，其表达式为

$$\hat{d}_{\mathrm{M}}=\sum_{|d_n-d_m|\leqslant\delta}d_n\sigma(-C_n),\tag{2}$$

式中：$\hat{d}_{\mathrm{M}}$ 为视差预测值；$d_m=D_{\max}m/N_{\mathrm{d}}$，$m=\underset{0\leqslant n\leqslant N_{\mathrm{d}}}{\operatorname{argmax}}(-C_n)$，实验中取 $\delta=2$。

为使(2) 式得到更好的视差估计，结合亚像素的交叉熵损失 L^{CE}和平滑的 L1 损失 L_1^{s} 训练模型，损失函数为

$$L=L^{\mathrm{CE}}+wL_1^{\mathrm{s}},\tag{3}$$

式中：w 为 L_1^{s} 的权重，用于平衡两种损失函数的重要性，实验中取 0.1；L^{CE}和 L_1^{s} 的具体公式为

$$L^{\mathrm{CE}}=\frac{1}{N}\sum_{i=0}^{N}\sum_{n=0}^{N_{\mathrm{d}}}-Q(d^{\mathrm{gt}},d_n)\ln[\sigma(-C_n)],\tag{4}$$

$$L_1^{\mathrm{s}}=\frac{1}{N}\sum_{i=0}^{N}f_{\mathrm{SL}}(|d^{\mathrm{gt}}-\hat{d}_{\mathrm{A}}|),\tag{5}$$

式中：N 为具有视差标签值的像素数；i 为像素索引；$Q(d^{\mathrm{gt}},d_n)=\exp(-|d_n-d^{\mathrm{gt}}|/b)$，为目标概率分布，是以视差标签值 d^{gt}为中心、散度为 b 的拉普拉斯分布，实验中取 $b=2$；$f_{\mathrm{SL}}(x)=\begin{cases}0.5x^2, & |x|<1\\ |x|-0.5, & \text{otherwise}\end{cases}$。

4　实　　验

在 SceneFlow 数据集[11]、KITTI2015 数据集[17]和 KITTI2012 数据集[18]上，使用评价指标 E_{EP}和 E_{D1}对算法进行评价。其中，E_{EP}表示预测视差与真值之间的差值绝对值；E_{D1}表示每组图像对评价区域的错误像素百分比，其中 E_{EP}小于 3 pixel 或 E_{EP}小于真值 5%时，认为是正确像素，否则为错误像素。

4.1　实验细节

所提算法使用 PyTorch 实现，源代码见 https://www.github.com/Wyf_2017/WSMCnet，所有训练和测试均在 2 个 Nvidia 1070ti 显卡上运行。使用小批量随机梯度下降的方式进行训练，单次迭代的样本大小取 2，4，8(保证训练时不会出现内存溢出的情况下取最大值)，梯度单次更新的迭代次数取 8，4，2，使单次梯度更新的样本大小扩展为 16。模型的训练均使用 Adam 优化器[27]，延迟率参数取(0.9，0.999)，图片随机裁剪的大小为 256 pixel×512 pixel，最大视差设为 192 pixel。使用的数据集如下：

1) SceneFlow 数据集：一个大型的合成数据集，包含 35454 对训练图像(记为 SF-train)和 4370 对测试图像(记为 SF-test)。每对图像的像素大小为 540 pixel×960 pixel，可以提供密集精细的视差图真值。实验中计算损失和评价指标时，排除视差大于 192 pixel 的图像。

2) KITTI2015 数据集和 KITTI2012 数据集：均为在不同天气条件下对街区真实场景记录的数据集，KITTI2015 数据集包含 200 对训练图像(记为 K15-train)和 200 对测试图像(记为 K15-test)，KITTI2012 数据集包含 194 对训练图像(记为 K12-train)和 195 对测试图像(记为 K12-test)，只使用其中的彩色图像对。每对图像的像素大小为375 pixel×1242 pixel，可以提供稀疏的激光雷达数据作为视差图的真值。为了对不同设置进行分析，将两个数据集中的训练图像的前 160 对图像作为训练集(记为 K-train)，其余的作为验证集(记为 K-val)。

为提高模型的泛化能力，对训练数据进行颜色增强和空间变换增强。其中，颜色增强包括色调增强、对比度增强、亮度增强和随机灰度化；为保证双目图像的核线几何特性，空间变换增强只包括随机裁剪和随机水平翻转。

4.2　超参数分析

对超参数的分析主要分为 3 组，前 2 组分别对

3.1 节中的 S 值和 C 值的作用进行分析，仅使用 SF-train 训练模型；第 3 组对两种损失函数进行对比分析，使用 SF-train 和 K-train 训练模型。其中，在 SF-train 上的实验，均先以学习率为 0.001 训练 15 个 epoch(1 个 epoch 就是将模型在数据集的所有样本上训练一次)，再以学习率为 0.0001 训练 5 个 epoch；在 K-train 上的实验，均先以学习率为 0.0005 训练 450 个 epoch，再以学习率为 0.0001 训练 150 个 epoch。

4.2.1　S 值对算法性能的影响

分析 S 值的同时，为与基准算法进行直接对比，本组实验在文献[14]的模型设置(其中 $C=1$)基础上，设定 S 的取值范围为[1, 8]，其性能对比如表 1 所示。表中 GPU 表示对于 256 pixel×256 pixel 的图片在训练阶段占用的 GPU 显存，E_{EP} 和 E_{D1} 均为训练完成时在验证集上的测试结果，t_{Run} 为 20 个随机样本(每个样本的图片大小为 540 pixel×960 pixel)上测试的平均值。可以看出，随着 S 值增大，算法误差逐渐增大，其计算负载逐渐降低。与文献[14]相比，同样模型设置的情况下，本实验中 E_{EP} 误差减小了约 6%，这主要得益于训练周期的延长和学习率的适当调整。与 $S=1$ 相比，$S=2$，3 时 E_{EP} 增加了约 8%，14%，E_{D1} 增加了 0.58%，0.93%，t_{Run} 降低了约 40%，53%；而与文献[14]相比，$S=2$，3 时 E_{EP} 仅增加了约 1%，6%，误差的增加较小，而运行时间显著缩短。

表 1　不同 S 值设置下算法的性能评价($C=1$)

Table 1　Performance evaluation of proposed method with different S ($C=1$)

Method	S	E_{EP}/pixel	E_{D1}/%	t_{Run}/s	GPU /GB
PSMNet[14]	1	1.09	—	—	—
Proposed	1	**1.02**	**3.41**	0.75	2.16
	2	1.10	3.89	0.45	1.51
	3	1.16	4.34	0.35	1.32
	4	1.22	4.81	0.30	1.20
	5	1.30	5.25	0.25	1.09
	6	1.34	5.62	0.25	1.08
	7	1.41	6.07	0.22	1.01
	8	1.42	6.23	**0.22**	**1.01**

4.2.2　C 值对算法性能的影响

从上述分析可知，当 $S=2$，3 时算法性能并不会有明显地降低，而运行效率得到了显著提高。因此，设定 $S=2$，3 时，C 的取值范围为[1, 6]，算法的性能对比如表 2 所示。可以看出，当 S 值固定时，算法的计算负载随 C 值的增大而略有增加，C 值的范围为[1, 3]时，误差随 C 值的增大而减小，当 C 值的范围为[4, 6]时，并不能维持这种变化趋势，这主要是因为较大的 C 值会使模型的训练难度加大。与 $S=1$，$C=1$ 相比，当 $S=2$，3，$C=3$ 时，E_{EP} 增加了约 3%，9%，E_{D1} 增加了 0.22%，0.58%，t_{Run} 降低了约 33%，48%；而与文献[14]相比，当 $S=2$，$C=3$ 时，E_{EP} 降低了约 4%，t_{Run} 降低了约 33%，算法精度和效率均得到提升；当 $S=3$，$C=3$ 时，E_{EP} 仅增加了约 2%，t_{Run} 降低了约 48%，损失较小精度的情况下效率得到了显著提高。

表 2　不同 C 值设置下算法的性能评价

Table 2　Performance evaluation of proposed method with different C

S	C	E_{EP}/pixel	E_{D1}/%	t_{Run}/s	GPU /GB
1	1	**1.02**	**3.41**	0.75	2.16
2	1	1.10	3.89	0.45	1.51
	2	1.07	3.71	0.48	1.68
	3	1.05	3.63	0.51	1.85
	4	1.08	3.81	0.53	2.02
	5	1.08	3.79	0.54	2.20
	6	1.07	3.69	0.56	2.37
3	1	1.16	4.34	**0.35**	**1.32**
	2	1.13	4.06	0.37	1.42
	3	1.11	3.99	0.39	1.55
	4	1.14	4.06	0.41	1.66
	5	1.14	4.03	0.42	1.78
	6	1.09	3.89	0.43	1.89

4.2.3　视差回归函数对算法性能的影响

从上述分析可知，与 $S=1$，$C=1$ 相比，当 $S=2$，3，$C=3$ 时，误差增加了约 3%，9%，而效率显著提高了约 33%，48%。因此，设定 $S=2$，3，$C=3$，研究两种损失函数对算法的影响，其性能对比如表 3 所示。表中 L1 表示使用(1)式和平滑的 L1 损失来训练模型，CE 表示使用(2)式和(3)式中的交叉熵损失来训练模型，CE+L1 表示使用(2)式和(3)式来训练模型，Tri/Bi 表示对匹配损失在 3/2 个维度上进行线性上采样。可以看出，在训练模型时，与使用平滑的 L1 损失相比，采用交叉熵损失对评价指标 E_{D1} 的改善较为明显，且当直接扩展视差范围时算法精度的降幅更小。与参数设置为{1，1，L1，Tri}相比，参数设置为{2，3，CE+L1，Bi}和

表 3　不同模型设置下算法的性能评价

Table 3　Performance evaluation of proposed method with different settings

Setting				Max disparity of 192				Max disparity of 384	
S	C	Loss	Dimensionality	E_{EP}/pixel	E_{D1}/%	t_{Run}/s	GPU /GB	E_{EP}/pixel	E_{D1}/%
1	1	L1	Tri	**1.02**	3.41	0.75	2.16	1.33	3.64
2	3	L1	Tri	1.05	3.63	0.51	1.85	1.38	3.90
2	3	CE	Bi	1.04	2.71	0.45	1.50	1.29	2.92
2	3	CE+L1	Bi	1.04	**2.69**	0.45	1.56	**1.28**	**2.87**
3	3	L1	Tri	1.11	3.99	0.39	1.55	1.49	4.30
3	3	CE	Bi	1.13	2.73	0.36	1.30	1.39	3.40
3	3	CE+L1	Bi	1.12	2.75	**0.36**	**1.28**	1.37	3.00

{3, 3, CE+L1, Bi}时，E_{EP}分别增加了约 2%和 10%，E_{D1}分别降低了 0.72%和 0.66%，t_{Run}分别降低了约 40%和 52%，这不仅减少了异常值的像素数，而且大幅度提高了运行效率。

选择表 3 中的三组模型设置 S_1={1, 1, L1, Tri}，S_2={2, 3, CE+L1, Bi}，S_3={3, 3, CE+L1, Bi}，在 K-train 上进行微调，在 K-val 上进行测试的性能对比如表 4 所示，表中 K15-val 和 K12-val 分别对应 K-val 中的两个子集。可以看出，与基准算法相比，所提算法不仅提高了算法运行效率，而且以 E_{D1}为评价指标时具有一定优势。

表 4　在 K-val 上不同参数设置下算法的性能评价

Table 4　Performance evaluation of proposed method with different settings on K-val

Setting	K15-val		K12-val	
	E_{EP}/pixel	E_{D1}/%	E_{EP}/pixel	E_{D1}/%
S_1	**0.74**	2.23	**0.62**	2.05
S_2	0.75	**2.02**	0.63	**1.78**
S_3	0.81	2.23	0.70	1.98

4.3　几种典型算法的性能对比

对 K15-test 和 K12-test 中的图像对使用所提算法（表 4 中参数设置 S_2）进行视差预测，并将其结果提交到 KITTI 数据集用于在线评价服务。基于深度学习的几种典型算法的性能对比如表 5 和表 6 所示，其中“All”表示评价时包含所有像素，“Noc”表示只考虑非遮挡区域内的像素。表 5 中 E_{D1-bg}、E_{D1-fg}和 E_{D1-all}分别表示在背景区域、前景区域和所有区域内计算评价指标 E_{D1}；表 6 中 γ_n表示评价区域内，E_{EP}大于 n 的像素百分比。可以看出，与几种典型算法相比，所提算法在精度和运行效率上均具有一定优势，且在 K15-test 和 K12-test 的非遮挡区域取得了最高的精度。

为分析所提算法存在的主要问题，在 K15-test 和 K12-test 上，分别选择误差较大的 2 组图像进行主观评价，结果如图 3 所示。其中，误差图为预测视差图与基准值之间的差值绝对值，左侧 2 组来自 K15-test（E_{D1-all}分别为 3.80%和 3.55%），右侧 2 组来自 K12-test（γ_3分别为 3.80%和 3.55%）。可以看出，在 4 种场景下所提算法均能得到合理且稠密的视差图，算法能够很好地处理纹理重复区域、弱/无纹理区域，如场景中道路、天空和车辆等所在的区域。但对于不规则表面、被遮挡和光线特别暗的区域仍存在较大的误差，如草地、物体边缘和栅栏等所在区域。

表 5　K15-test 上不同算法的性能评价

Table 5　Performance evaluation of different methods on K15-test

Method	All			Noc			t_{Run}/s
	E_{D1-bg}/%	E_{D1-fg}/%	E_{D1-all}/%	E_{D1-bg}/%	E_{D1-fg}/%	E_{D1-all}/%	
MC-CNN-arct[10]	2.89	8.88	3.89	2.48	7.64	3.33	67.00
DispNetC[11]	4.32	4.41	4.34	4.11	3.72	4.05	**0.06**
iResNet-i2[12]	2.25	3.40	2.44	2.07	2.76	2.19	0.12
GC-net[13]	2.21	6.16	2.87	2.02	5.58	2.61	0.90
PSMNet[14]	1.86	4.62	2.32	1.71	4.31	2.14	0.41
Proposed	**1.72**	**4.19**	**2.13**	**1.51**	**3.57**	**1.85**	0.39

表 6　K12-test 上不同算法的性能评价

Table 6　Performance evaluation of different methods on K12-test

Method	γ_2 / %		γ_3 / %		γ_5 / %		Mean E_{EP} / pixel	
	Noc	All	Noc	All	Noc	All	Noc	All
MC-CNN-arct[10]	3.90	5.45	2.43	3.63	1.64	2.39	0.7	0.9
DispNetC[11]	7.38	8.11	4.11	4.65	2.05	2.39	0.9	1.0
iResNet-i2[12]	2.69	3.34	1.71	2.16	1.06	1.32	0.5	0.6
GC-net[13]	2.71	3.46	1.77	2.30	1.12	1.46	0.6	0.7
PSMNet[14]	2.44	**3.01**	1.49	**1.89**	0.90	**1.15**	**0.5**	**0.6**
Proposed	**2.35**	3.04	**1.42**	1.90	**0.89**	1.19	0.6	0.6

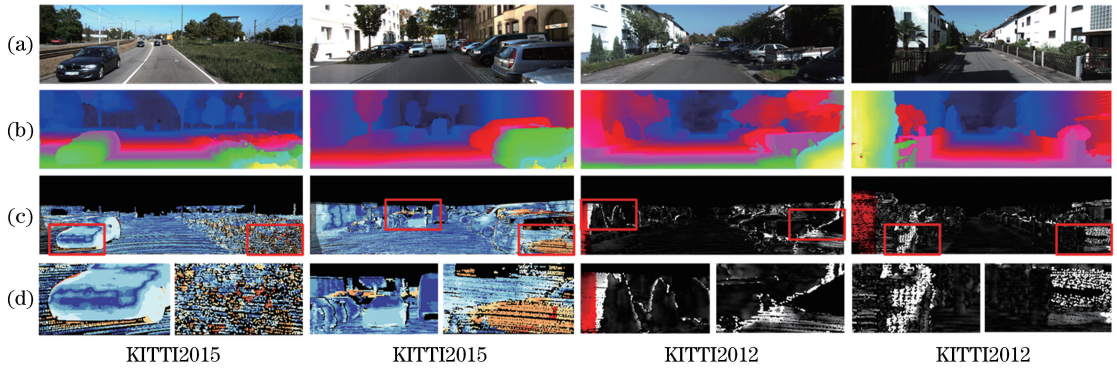

图 3　算法预测的视差结果。(a)左视角图像;(b)视差图;(c)误差图;(d)局部细节

Fig. 3　Disparity predicted by proposed method. (a) Left image; (b) disparity map; (c) error map; (d) local details

5　结　　论

提出一种在视差维度上使用稀疏损失体进行立体匹配的方法,在 K15-test 和 K12-test 上,与几种典型的基于深度学习的方法相比,所提算法在整体精度上取得了最优性能,特别是与基准方法相比,不仅提高了算法精度,而且运行时间也大幅度缩短。但对于小型设备算法开销较大,仍无法满足实时性要求,为得到精度高、运行效率高的立体匹配算法,仍需要对算法的网络结构进行更深入的研究。

参　考　文　献

[1] Nguyen V D, Nguyen D D, Lee S J, *et al*. Local density encoding for robust stereo matching [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2014, 24(12): 2049-2062.

[2] Heise P, Jensen B, Klose S, *et al*. Fast dense stereo correspondences by binary locality sensitive hashing [C]//2015 IEEE International Conference on Robotics and Automation (ICRA), May 26-30, 2015, Seattle, WA, USA. New York: IEEE, 2015: 105-110.

[3] Taniai T, Matsushita Y, Sato Y, *et al*. Continuous 3D label stereo matching using local expansion moves [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 40(11): 2725-2739.

[4] Hamzah R A, Ibrahim H, Hassan A H A. Stereo matching algorithm based on per pixel difference adjustment, iterative guided filter and graph segmentation[J]. Journal of Visual Communication and Image Representation, 2017, 42: 145-160.

[5] He K M, Zhang X Y, Ren S Q, *et al*. Identity mappings in deep residual networks[M]//Leibe B, Matas J, Sebe N, *et al*. European conference on computer vision-ECCV 2016. Lecture notes in computer science. Cham: Springer, 2016, 9908: 630-645.

[6] Ren S Q, He K M, Girshick R, *et al*. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[7] Shelhamer E, Long J, Darrell T. Fully convolutional networks for semantic segmentation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(4): 640-651.

[8] Xiao J S, Tian H, Zou W T, *et al*. Stereo matching based on convolutional neural network [J]. Acta

Optica Sinica, 2018, 38(8): 0815017.
肖进胜, 田红, 邹文涛, 等. 基于深度卷积神经网络的双目立体视觉匹配算法[J]. 光学学报, 2018, 38(8): 0815017.

[9] Žbontar J, LeCun Y. Computing the stereo matching cost with a convolutional neural network[C]//2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 7-12, 2015, Boston, MA, USA. New York: IEEE, 2015: 1592-1599.

[10] Žbontar J, LeCun Y. Stereo matching by training a convolutional neural network to compare image patches[J]. Journal of Machine Learning Research, 2016, 17(1): 2287-2318.

[11] Mayer N, Ilg E, Häusser P, *et al*. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 4040-4048.

[12] Pang J H, Sun W X, Ren J S, *et al*. Cascade residual learning: a two-stage convolutional neural network for stereo matching[C]//2017 IEEE International Conference on Computer Vision Workshops (ICCVW), October 22-29, 2017, Venice, Italy. New York: IEEE, 2017: 878-886.

[13] Kendall A, Martirosyan H, Dasgupta S, *et al*. End-to-end learning of geometry and context for deep stereo regression[C]//2017 IEEE International Conference on Computer Vision (ICCV), October 22-29, 2017, Venice, Italy. New York: IEEE, 2017: 66-75.

[14] Chang J R, Chen Y S. Pyramid stereo matching network[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE, 2018: 5410-5418.

[15] Scharstein D, Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms[J]. International Journal of Computer Vision, 2002, 47(1/2/3): 7-42.

[16] Scharstein D, Hirschmüller H, Kitajima Y, *et al*. High-resolution stereo datasets with subpixel-accurate ground truth[M]//Jiang X, Hornegger J, Koch R. German conference on pattern recognition. GCPR 2014. Lecture notes in computer science. Cham: Springer, 2014, 8753: 31-42.

[17] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite[C]//2012 IEEE Conference on Computer Vision and Pattern Recognition, June 16-21, 2012, Providence, RI, USA. New York: IEEE, 2012: 3354-3361.

[18] Menze M, Geiger A. Object scene flow for autonomous vehicles[C]//2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 7-12, 2015, Boston, MA, USA. New York: IEEE, 2015: 3061-3070.

[19] Luo W J, Schwing A G, Urtasun R. Efficient deep learning for stereo matching[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE, 2016: 5695-5703.

[20] Liang Z F, Feng Y L, Guo Y L, *et al*. Learning for disparity estimation through feature constancy[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE, 2018: 2811-2820.

[21] Jie Z Q, Wang P F, Ling Y G, *et al*. Left-right comparative recurrent model for stereo matching[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 18-23, 2018, Salt Lake City, UT, USA. New York: IEEE, 2018: 3838-3846.

[22] Yu L D, Wang Y C, Wu Y W, *et al*. Deep stereo matching with explicit cost aggregation sub-architecture[J/OL]. (2018-01-12)[2019-04-15]. http://cn.arxiv.org/abs/1801.04065.

[23] Smolyanskiy N, Kamenev A, Birchfield S. On the importance of stereo for accurate depth estimation: an efficient semi-supervised deep neural network approach[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 18-22, 2018, Salt Lake City, UT, USA. New York: IEEE, 2018: 1120-1128.

[24] Zhong Y R, Dai Y C, Li H D. Self-supervised learning for stereo matching with self-improving ability[J/OL]. (2017-09-04)[2019-04-15]. http://cn.arxiv.org/abs/1709.00930.

[25] Wang Y F, Wang H W, Wu C, *et al*. Self-supervised stereo matching algorithm based on common view[J]. Acta Optica Sinica, 2019, 39(2): 0215004.
王玉锋, 王宏伟, 吴晨, 等. 基于共同视域的自监督立体匹配算法[J]. 光学学报, 2019, 39(2): 0215004.

[26] Girshick R. Fast R-CNN[C]//2015 IEEE International Conference on Computer Vision (ICCV), December 7-13, 2015, Santiago, Chile. New York: IEEE, 2015: 1440-1448.

[27] Kingma D P, Ba J. Adam: a method for stochastic optimization[J/OL]. (2017-01-30)[2019-04-15]. http://cn.arxiv.org/abs/1412.6980.