

基于高斯拉普拉斯算子的加权引导图滤波 立体匹配算法

周博^{1,2*}, 秦岭^{1,2**}, 龚伟^{1,2***}

¹现代汽车零部件技术湖北省重点实验室(武汉理工大学), 湖北 武汉 430070;

²现代汽车零部件技术湖北省协同创新中心, 湖北 武汉 430070

摘要 提出了一种基于高斯拉普拉斯(LoG)算子的加权引导图滤波立体匹配算法。采用加权绝对差与梯度相融合的方式计算匹配代价。利用基于 LoG 算子的引导图滤波进行代价聚合,使惩罚参数能够随图像中的不同纹理区域进行自适应调整。使用赢家通吃(WTA)策略计算视差,并采用二次插值法得到最终的视差图。对 Middlebury 平台中的图像进行测试,结果表明,所提算法的平均误匹配率为 4.32%。该算法可以有效地兼顾图像的高纹理区域和低纹理区域,并降低了视差图的误匹配率。

关键词 机器视觉; 立体匹配; 加权绝对差; 高斯拉普拉斯算子; 引导图滤波

中图分类号 TN911.73

文献标识码 A

doi: 10.3788/LOP56.101502

Stereo-Matching Algorithm Using Weighted Guided Image Filtering Based on Laplacian of Gaussian Operator

Zhou Bo^{1,2*}, Qin Ling^{1,2**}, Gong Wei^{1,2***}

¹Hubei Key Laboratory of Advanced Technology for Automotive Components (Wuhan University of Technology),
Wuhan, Hubei 430070, China;

²Hubei Collaborative Innovation Center for Automotive Components Technology, Wuhan, Hubei 430070, China

Abstract A stereo-matching algorithm using weighted guided image filtering based on the Laplacian of Gaussian (LoG) operator is proposed. The algorithm calculates matching cost by fusing weighted absolute difference and gradient. Then, cost aggregation is implemented using an improved guided image filtering based on the LoG operator to ensure that the penalty parameter is self-adaptive. The disparity computation is implemented using the winner-take-all (WTA) strategy, and the final disparity map is obtained using two different interpolation methods. The experimental results show that the average mismatch rate of the proposed algorithm on the Middlebury benchmark standard dataset is 4.32%. The proposed algorithm can process both high and low texture regions effectively; thus, the mismatch rate of the disparity map is reduced.

Key words machine vision; stereo matching; weighted absolute difference; Laplacian of Gaussian operator; guided image filtering

OCIS codes 150.1135; 330.1400; 100.6890

1 引言

立体匹配算法利用同一场景的两幅图像来计算场景的深度信息。根据计算过程的不同,立体匹配算法大致可分为两类:全局立体匹配算法和局部立

体匹配算法。全局立体匹配算法的核心是构建全局能量函数,通过最小化全局能量函数得到最优视差值。常见的全局立体匹配算法有:动态规划^[1]、图割法^[2]、置信传播^[3]等,其结果不但精度高而且效果好,但是计算复杂度高、效率低。局部立体匹配算法

收稿日期: 2018-11-27; 修回日期: 2018-12-08; 录用日期: 2018-12-14

基金项目: 国家重点研发计划(2018YFC0808405)

* E-mail: zhoubo180@foxmail.com; ** E-mail: qinling@whut.edu.cn; *** E-mail: gongwei_whut@163.com

一般基于窗口进行代价聚合,采用赢者通吃(WTA)策略进行视差选择,其结果的精度一般低于全局立体匹配算法,但是计算复杂度低、效率高,在实际应用中更加易于实现。

在早期的立体匹配研究中,研究者往往采用单一的相似性测度来进行代价计算,其结果易受到环境变化的影响。Mei等^[4]提出绝对差(AD)与Census变换相结合的方式,同时考虑了图像局部结构相似区域及图像色彩分布相似区域,使代价聚合具有较强的稳健性。卢迪等^[5]结合了图像Census变换、Weber局部描述符(WLD)特征、AD和梯度4种相似性测度来进行代价计算,并得到较好的匹配结果,但匹配代价的计算较为复杂。范海瑞等^[6]将改进的Census变换和梯度进行融合,从而提高了匹配代价对噪声的抗干扰能力。针对单一相似性测度易受环境变化的影响的问题,采用多种相似性测度相结合的方式是一个很好的解决办法,在实际中也能得到广泛的应用。

传统的局部立体匹配算法在进行代价聚合时多采用固定窗口的方式,这种方式简单有效,但是在深度不连续区域,其匹配效果很差。Zhang等^[7]提出了一种十字交叉的自适应窗口进行代价聚合,取得了良好的匹配效果,但该方法在斜面区域和一些噪声影响的实际场景中表现不佳。Yoon等^[8]提出了一种自适应权重的算法,该算法需要计算矩形窗口内各像素点的权值,运行速度较慢。基于窗口的代价聚合算法无法避免窗口尺寸变化对立体匹配结果的影响。Rhemann等^[9]使用引导图滤波的方法来进行代价聚合,该方法的计算复杂度与窗口的尺寸无关,计算量也较小。不足之处是Rhemann所使用的传统引导图滤波采用了固定的惩戒参数,这导致

在图像边缘附近会不可避免地产生光晕现象,影响立体匹配的精确度。

本文针对上述问题,使用AD、梯度两种相似性测度相结合的方式来进行初始代价计算,在代价聚合阶段,采用基于高斯拉普拉斯(LoG)算子进行加权的改进引导图滤波,从而实现了高效、高精度的立体匹配。

2 算法描述

2.1 代价计算

使用AD作为匹配代价的算法多采用单通道AD或者均值AD,其中均值AD是先分别计算红、绿、蓝(R、G、B)三通道内AD,然后再求三者的平均值。为充分保留原始图像的色彩信息,本文采用加权的AD计算方式,为R、G、B三通道分别分配不同的权值,通过加权求和得到AD匹配代价。定义 p 为左图像的像素点, q 为对应右图像的像素点, $C_{AD}=(p,d)$ 为 p 在视差值为 d 时的AD匹配代价。在计算初始代价时添加截断值 T_1 ,则

$$C_{AD}(p,d) = \min \left[\sum_{c=R,G,B} A_c |I_{c,Left}(p) - I_{c,Right}(q)|, T_1 \right], \quad (1)$$

式中: c 为R、G、B计算通道; $I_{c,Left}(p)$ 、 $I_{c,Right}(q)$ 分别为左、右图像在通道 c 中,点 p 、 q 的像素值; A_c 为AD在 c 通道的加权项,其中 $A_R=0.299$, $A_G=0.587$, $A_B=0.114$ 。

加权AD作为匹配代价能够更好地保持原始图像的颜色信息。从图1能够看出,采用加权AD作为匹配代价生成视差图的效果明显好于采用单通道AD或均值AD的视差图。



图1 不同匹配代价对比。(a)单通道AD;(b)均值AD;(c)加权AD

Fig. 1 Comparison of different matching costs. (a) Single channel AD; (b) mean AD; (c) weighted AD

为增强代价计算阶段的稳健性,采用AD与梯度相结合的方式来进行代价计算。计算图像水平方向的梯度,将该梯度与一个截断值进行比较得到梯度匹配代价。定义 G_x 为图像 x 方向(即水平方向)梯度、 $C_{grad}(p,d)$ 为像素点 p 在视差值为 d 时的梯度匹配代价,并在计算时添加截断值 T_2 ,则

$$C_{grad}(p,d) = \min(G_x, T_2). \quad (2)$$

结合AD匹配代价和梯度匹配代价即可得到最终匹配代价,即

$$C(p,d) = \alpha \cdot C_{AD}(p,d) + (1-\alpha) \cdot C_{grad}(p,d), \quad (3)$$

式中: α 为AD匹配代价所占权重。

2.2 代价聚合

引导图滤波算法^[10]一般是基于一个局部线性模型实现的,设引导图滤波模型为

$$q_i = a_k I_i + b_k, \quad (4)$$

式中: k 为窗口 ω_k 内的中心像素; i 为图像某窗口 ω_k 内的像素; a_k 、 b_k 为窗口中的线性系数; I_i 为与像素 i 对应的引导图像; q_i 为与像素 i 对应的滤波输出的图像。令 p_i 为与像素 i 对应的输入图像,则 a_k 、 b_k 的值可通过最小化 q 、 p 的差值来确定。一般地,引导图滤波算法构建的优化函数模型为

$$E(a_k, b_k) = \sum_{i \in \omega_k} [(a_k I_i + b_k - p_i)^2 + \epsilon a_k^2], \quad (5)$$

式中: E 为函数模型能量; ϵ 为惩戒参数,能够防止 a_k 取值过大。通过最小二乘法可求得窗口系数为

$$a_k = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_k} p_i I_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon}, \quad (6)$$

$$b_k = \bar{p}_k - a_k \mu_k, \quad (7)$$

式中: $|\omega|$ 为窗口 ω_k 中的像素总数; μ_k 、 σ_k 分别为引导图在窗口 ω_k 内的均值和标准差; \bar{p}_k 为输入图像在窗口 ω_k 内的均值。

使用传统引导图滤波进行代价聚合时,为防止 a_k 过大,一般会采用一个小于1的固定惩戒参数 ϵ 。当使用引导图滤波对输入图像进行处理时,为了更好地保留输入图像的纹理信息,较为理想的处理方法是:在图像的边缘位置,采用较小的平滑倍数,即惩戒参数 ϵ 应较小;在图像的低纹理区域,采用较大的平滑倍数,即惩戒参数 ϵ 应较大。基于此,本文提出了一种基于LoG算子的加权引导图滤波改进算法。利用基于LoG算子的加权项来检测图像纹理,从而使惩戒参数能够自适应调整,并兼顾图像中不同的纹理区域。

LoG算子的表达式为

$$\text{LoG}(x, y) = \left(\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2} \right), \quad (8)$$

式中: x 、 y 分别为像素点的列、行坐标; σ 为标准差。通过零交叉点检测,LoG算子能够有效地识别出图像的边缘。

改进的引导图滤波算法通过计算图像局部窗口内的LoG边缘来判断像素点是处于低纹理区域还是高纹理区域。基于此边缘信息,在惩戒参数

的分母中引入一个权值项 $T(s)$,并利用一个指数函数来增强权值项的作用效果。权值项 $T(s)$ 的表达式为

$$T(s) = \frac{1}{N} \sum_{i=1}^N \frac{|\text{LoG}(s)| + \delta}{|\text{LoG}(s')| + \delta}, \quad (9)$$

式中: s 为局部窗口内的中心像素; s' 为局部窗口内的所有像素; $|\text{LoG}(s)|$ 表示高斯拉普拉斯算子幅值; N 为局部窗口内的像素总数; δ 为窗口内高斯拉普拉斯算子幅值最大值的十分之一。因为图像边缘像素点的LoG算子幅值总是大于图像低纹理区域像素点的LoG算子幅值,所以权值项 $T(s)$ 在边缘处总是大于1,而在低纹理区域则小于1。最终(5)式转化为

$$E(a_k, b_k) = \sum_{i \in \omega_k} \left\{ (a_k I_i + b_k - p_i)^2 + \frac{\epsilon}{\exp\left[\frac{T(s)}{\gamma}\right] - 1} a_k^2 \right\}, \quad (10)$$

式中: γ 为一个小于1的参数。

使用最小二乘法可得到改进的引导图滤波窗口线性系数分别为

$$a'_k = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_k} p_i I_i - \mu_k \bar{p}_k}{\sigma_k^2 + \frac{\epsilon}{\exp\left[\frac{T(s)}{\gamma}\right] - 1}}, \quad (11)$$

$$b'_k = \bar{p}_k - a'_k \mu_k. \quad (12)$$

分别使用改进前的引导图滤波和改进后的引导图滤波进行代价聚合,得到对应的左视差图如图2所示。图2中的分图均未进行精细化处理,图中的红色点为非遮挡区域的误匹配点,设置误差限为1 pixel,即当计算得到的视差图与真实视差图在同一位置视差值的差值大于1时,该点即被判断为误匹配点。误匹配率为误匹配点的个数占视差图中所有像素点个数的百分比。分别计算图2中的测试图集在引导图滤波改进前、后进行代价聚合所得的视差图在非遮挡区域的误匹配率,结果如图3所示。从图2可以看出,使用改进前的引导图滤波进行代价聚合所得的视差图在非遮挡区域中误匹配点数要明显多于使用改进后算法所得视差图的误匹配点数,采用基于LoG算子加权的改进引导图滤波有效地保留了图像的边缘信息。图3直观地显示出使用改进后的引导图滤波算法进行代价聚合的方法有效地降低了视差图在非遮挡区域的误匹配率。

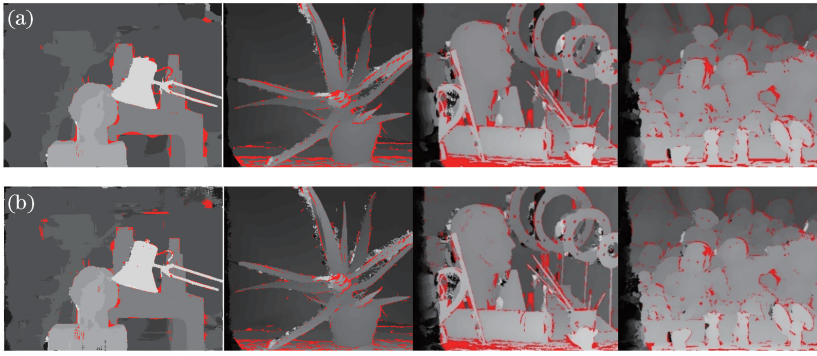


图 2 引导图滤波改进前、后视差图的对比。(a)改进前;(b)改进后

Fig. 2 Comparison of disparity maps before and after improvement by guided image filtering.

(a) Before improvement; (b) after improvement

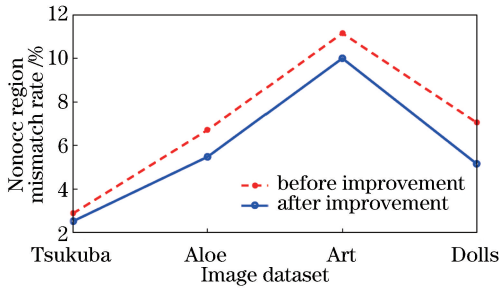


图 3 引导图滤波改进前、后误匹配率的对比

Fig. 3 Comparison of mismatch rates before and after improvement by guided image filtering

2.3 视差选择与精细化

WTA 策略是一种简单有效的视差选择方法。该策略通过选择同一像素点在不同视差空间中的最小值来快速地生成原始视差图。

本文采用 WTA 策略进行视差选择。为了进一步提高视差图的准确率,还须对原始视差图进行精细化处理。在精细化阶段,对原始视差图进行左右一致性检测,检测表达式为

$$|d_{\text{Left}}(p) - d_{\text{Right}}[p - d_{\text{Left}}(p)]| \leq T, \quad (13)$$

式中: d_{Left} 为左图像在像素点 p 处的视差值; $d_{\text{Right}}[p - d_{\text{Left}}(p)]$ 为右图像在像素点 p 的对应点 $[p - d_{\text{Left}}(p)]$ 处的视差值; T 为设定的视差差异阈值,本文设置为 1 pixel。满足(13)式条件的像素点判断为正常匹配点,不满足则判断为异常匹配点。

针对异常匹配点,使用一种二次插值的方法处理。首先,针对左右一致性检测出的异常匹配点,分别沿视差图像的水平线向左、向右查找第一个正常匹配点,其对应的视差值分别记为 d'_L 、 d'_R 。令异常匹配点处的视差值为 d'_L 、 d'_R 中的最小值,即

$$d_p(u, v) = \min(d'_L, d'_R), \quad (14)$$

式中: $d_p(u, v)$ 为异常匹配点处的视差值, u, v 分别

表示左右一致性检测所得的异常匹配点在原始视差图中的列坐标、行坐标,下标 P 表示经过左右一致性检测后的视差图。此过程为第一次插值。

令经过第一次插值处理得到的视差图为 P_1 ,利用加权中值滤波对 P_1 进行处理得到视差图 P_2 。使用双边滤波函数来计算加权中值滤波的权值项,即

$$W_{m,n} = \frac{1}{K_m} \exp\left(-\frac{|m-n|^2}{\sigma_1^2}\right) \exp\left(-\frac{|I_m - I_n|^2}{\sigma_2^2}\right), \quad (15)$$

式中: σ_1, σ_2 分别用于调整空间和色彩相似度; K_m 为归一化参数; I_m, I_n 分别为双边滤波窗口内两个像素点 m, n 的像素值; $W_{m,n}$ 为权值项。

令视差图 P 在异常匹配点处的视差值为视差图 P_2 在同样位置的视差值,即

$$P(u, v) = P_2(u, v), \quad (16)$$

便可得到最终的视差图。此过程为第二次插值。使用 WTA 策略进行视差选择生成的视差图通过上述的二次插值处理后,得到了最终的视差图。

图 4 所示为视差选择与精细化过程中各阶段的图像。从图 4 可以看出,第一次插值后的视差图有着明显的条纹效应。为了消除条纹效应,提高视差图像的准确率,采用了上述的第二次插值方法。从图 5 可以看出,第二次插值有效地消除了图像中的条纹效应。此外,对于异常匹配点,第二次插值的结果要明显优于第一次插值的结果。

3 实验结果分析

为了验证所提算法的效果,采用 Middlebury 立体匹配算法测试平台提供的标准立体图像对该算法进行测试。所提算法是基于 Matlab 软件平台实现的,硬件平台中中央处理器为 Intel(R) Core(TM)

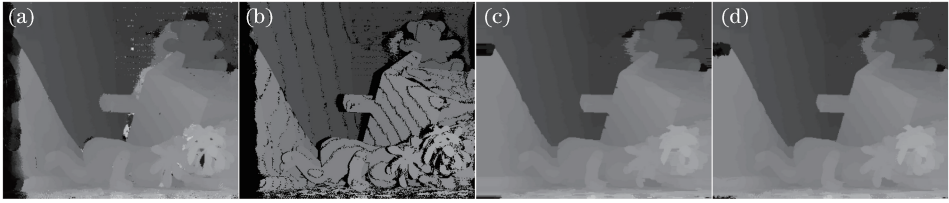


图4 视差选择与精细化过程。(a)左图像匹配结果;(b)左右一致性检测结果;(c)第一次插值结果;(d)第二次插值结果
Fig. 4 Disparity selection and refinement process. (a) Matching result of left image; (b) test result of left-right consistency; (c) result of first interpolation; (d) result of second interpolation

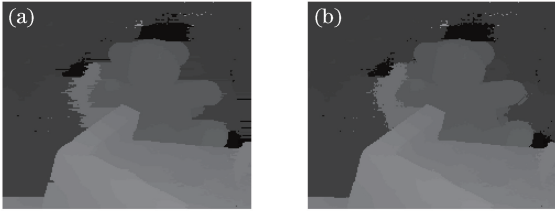


图5 二次插值细节对比。(a)第一次插值结果;
(b)第二次插值结果

Fig. 5 Comparison of details for two interpolations. (a) Result of first interpolation; (b) result of second interpolation

I5-8250U,主频为 1.60 GHz。为确保算法的稳定性,程序在处理不同图像时所用的参数均保持不变,具体参数为 $\{T_1, T_2, \alpha, \gamma\} = \{7/255, 3/255, 0.11, 0.25\}$ 。

使用 Middlebury 测试平台上的 4 组测试图集^[11](Tsukuba、Venus、Teddy 和 Cones)对算法的性能进行测试,结果如图 6 所示。为了更加直观地分析所提出的算法、量化算法性能之间的差异,计算立体匹配视差图在非遮挡区域和所有区域的误匹配率,并选取 6 种经典的立体匹配算法(GC-occ^[12]、RTCensus^[13]、Fast Aggreg^[14]、Semi Glob^[15]、Adapt Weight^[8]和 RTAdapt Weight^[16])进行比较。最终经精细化处理后的视差图误差限为 1 pixel,算法所得的结果如表 1 所示。从表 1 可以看出,在 4 组测试图集的最终结果中,所提算法的平均误匹配率为 4.32%,均低于其他算法。在 Venus 和 Cones 两组测试图集中,所提算法的效果明显优于其余算法,误匹配率表现最优。

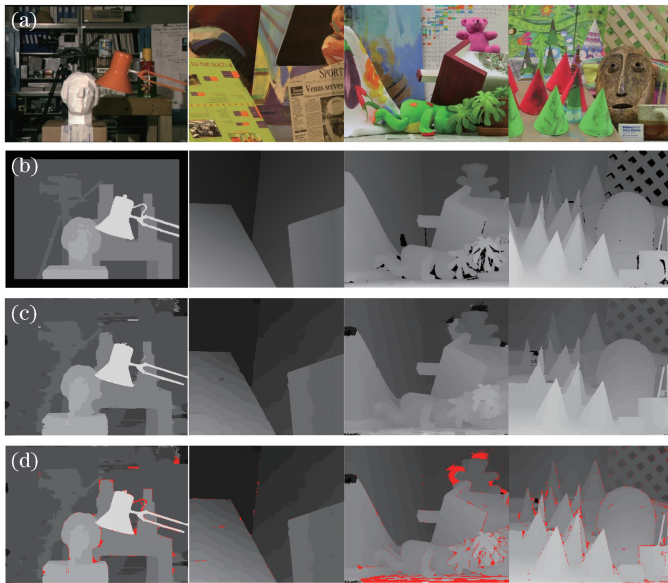


图6 所提算法的结果。(a)左测试图;(b)真实视差图;(c)所提算法获得的视差图;(d)所提算法获得的误匹配点
Fig. 6 Result of proposed algorithm. (a) Original left images; (b) ground truth; (c) disparity maps obtained by proposed algorithm; (d) mismatching points gained by proposed algorithm

使用 Middlebury 测试平台上其余 27 组测试图集^[17]对所提算法的性能进行进一步的测试。选取 3 种经典的算法(Adapt Weight^[8]、MST^[18]和 Fast

Cost^[9]),以非遮挡区域误匹配率作为对比项,分析算法的总体性能,测试结果如表 2 所示。表 2 中误匹配率的下标为算法的排序。所提算法在全部 27 组

表 1 不同立体匹配算法对 4 组测试图集非遮挡区域和全部区域的误匹配率

Table 1 Mismatching rates of different stereo-matching algorithms in non-occluded regions and all regions of four groups of test image sets

Algorithm	Tsukuba		Venus		Teddy		Cones		Average
	Nonocc	All	Nonocc	All	Nonocc	All	Nonocc	All	
GC-occ	1.19	2.01	1.64	2.19	11.20	17.40	5.36	12.40	6.67
RTCensus	5.08	6.25	1.58	2.42	7.96	13.80	4.10	9.54	6.34
Fast Aggreg	1.16	2.11	4.03	4.75	9.04	15.20	5.37	12.60	6.78
Semi Glob	3.26	3.96	1.00	1.57	6.02	12.20	3.06	9.75	5.10
Adapt Weight	1.38	1.85	0.71	1.19	7.88	13.30	3.97	9.79	5.01
RTAdapt Weight	1.45	1.99	0.40	0.81	7.65	13.30	3.48	9.34	4.80
Proposed algorithm	1.81	2.65	0.27	0.51	6.75	10.15	2.78	9.62	4.32

测试图集中有 17 组视差图的非遮挡区域误匹配率最低,平均误匹配率仅为 5.27%。Adapt Weight、MST 和 Fast Cost 三种算法所得的误匹配率分别为 8.39%、8.41%和 7.13%,远高于所提算法。在全部测试组中,所提算法所得的非遮挡区域误匹配率在其中的 5 组排名第二,平均误匹配率为 7.84%,仅高于 Fast Cost 算法的 5.54%,远低于 Adapt Weight 算法的 12.03%和 MST 算法的 13.54%。在剩余 5 个测试组中,所提算法的非遮挡区域误匹配率为 33.24%,均高于 Adapt Weight 算法的 27.47%、MST 算法的 20.10%和 Fast Cost 算法的 26.05%。分析所提算法的视差图与测试图集原图能够发现,当测试图像中相邻的目标与背景、目标与目标之间存在着极为相似的色彩信息,且这些色彩区域均为大范围的无纹理区域时,所提算法的立体匹配的误匹配率较高。

4 结 论

提出了一种基于 LoG 算子的加权引导图滤波立体匹配算法。在代价计算阶段,采用加权 AD 代价与梯度代价相结合的方式,有效保留了图像的色彩信息与结构信息,提高了匹配代价的可靠性。在代价聚合阶段,利用基于 LoG 算子的权值项改进了引导图滤波算法,使惩戒参数能够随图像中的不同纹理区域自适应变化。实验结果表明,改进后的引导图滤波算法不但能够有效地保留图像的纹理信息,而且能够降低视差图的误匹配率。在视差精细化阶段,使用二次插值的方法有效地去除了传统插值方法所带来的条纹效应,并进一步降低了图像的误匹配率。对比实验的结果表明,所提出的改进算法具有较好的性能。但是,从视差图误匹配点较多的区域可以看出,所提算法在处理一些细节区域时的匹配精度还有待提高。今后的研究将针对这一问

题对该算法进行优化。

表 2 不同立体匹配算法对其余 27 组测试图集非遮挡区域的误匹配率

Table 2 Mismatching rates of different stereo-matching algorithms in non-occluded regions of other 27 groups of test image sets

Stereo Dataset	Adapt Weight	MST	Fast Cost	Proposed algorithm
Aloe	6.50 ₂	6.61 ₃	7.06 ₄	5.16 ₁
Art	12.88 ₃	13.74 ₄	12.03 ₂	8.96 ₁
Baby1	5.45 ₃	7.65 ₄	3.16 ₁	3.60 ₂
Baby2	13.64 ₃	14.12 ₄	3.97 ₁	4.25 ₂
Baby3	8.76 ₃	9.53 ₄	4.71 ₁	6.09 ₂
Books	12.65 ₃	12.97 ₄	10.21 ₂	8.68 ₁
Bowling1	16.87 ₃	19.14 ₄	6.21 ₁	11.65 ₂
Bowling2	10.43 ₃	12.63 ₄	8.18 ₂	7.66 ₁
Cloth1	1.25 ₃	1.18 ₂	1.88 ₄	0.89 ₁
Cloth2	5.56 ₄	5.40 ₃	4.20 ₂	3.07 ₁
Cloth3	3.81 ₄	3.20 ₃	2.79 ₂	1.92 ₁
Cloth4	3.21 ₄	2.53 ₃	2.25 ₂	1.62 ₁
Dolls	6.43 ₂	7.86 ₄	7.12 ₃	4.51 ₁
Flowerpots	15.44 ₃	17.27 ₄	9.63 ₁	13.61 ₂
Lampshade1	12.36 ₄	11.29 ₂	12.12 ₃	10.80 ₁
Lampshade2	17.44 ₃	14.20 ₂	12.48 ₁	22.59 ₄
Laundry	16.89 ₂	19.39 ₃	20.64 ₄	13.53 ₁
Midd1	36.47 ₂	18.96 ₁	36.95 ₃	40.36 ₄
Midd2	35.56 ₃	20.21 ₁	34.02 ₂	38.08 ₄
Moebius	13.26 ₄	11.33 ₃	10.64 ₂	9.74 ₁
Monopoly	15.43 ₁	16.85 ₂	19.74 ₃	24.06 ₄
Plastic	32.44 ₃	30.30 ₂	27.06 ₁	41.09 ₄
Reindeer	11.73 ₃	11.96 ₄	7.84 ₂	4.94 ₁
Rocks1	5.43 ₄	4.14 ₃	3.97 ₂	2.24 ₁
Rocks2	5.02 ₄	3.33 ₃	2.67 ₂	1.39 ₁
Wood1	8.99 ₃	11.17 ₄	4.93 ₂	3.98 ₁
Wood2	6.15 ₄	4.20 ₃	2.71 ₂	0.48 ₁
Average	12.59 ₄	11.52 ₃	10.34 ₁	10.92 ₂

参 考 文 献

- [1] Xu J X, Li Q W, Liu Y, *et al.* Stereo matching algorithm based on color weights and tree dynamic programming[J]. *Acta Optica Sinica*, 2017, 37(12): 1215007.
许金鑫, 李庆武, 刘艳, 等. 基于色彩权值和树形动态规划的立体匹配算[J]. *光学学报*, 2017, 37(12): 1215007.
- [2] Zhang Q, Lu S Q, Li H B, *et al.* Research on underwater stereo matching method based on color segmentation[J]. *Acta Optica Sinica*, 2016, 36(8): 0815001.
张强, 卢士强, 李海滨, 等. 基于色彩分割的水下立体匹配算法的研究[J]. *光学学报*, 2016, 36(8): 0815001.
- [3] Wang X F, Su Y Y, Tang L M, *et al.* A combined back and foreground-based stereo matching algorithm using belief propagation and self-adapting dissimilarity measure [J]. *International Journal of Pattern Recognition and Artificial Intelligence*, 2018, 32(6): 1850019.
- [4] Mei X, Sun X, Zhou M C, *et al.* On building an accurate stereo matching system on graphics hardware[C] // 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011: 467-474.
- [5] Lu D, Lin X. Alocal stereo matching algorithm based on the combination of multiple similarity measures [J]. *Robot*, 2016, 38(1): 1-7.
卢迪, 林雪. 多种相似性测度结合的局部立体匹配算法[J]. *机器人*, 2016, 38(1): 1-7.
- [6] Fan H R, Yang F, Pan X R, *et al.* Stereo matching algorithm for improved Census transform and gradient fusion[J]. *Acta Optica Sinica*, 2018, 38(2): 0215006.
范海瑞, 杨帆, 潘旭冉, 等. 一种改进 Census 变换与梯度融合的立体匹配算法[J]. *光学学报*, 2018, 38(2): 0215006.
- [7] Zhang K, Lu J B, Lafruit G. Cross-based local stereo matching using orthogonal integral images[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2009, 19(7): 1073-1079.
- [8] Yoon K J, Kweon I S. Adaptive support-weight approach for correspondence search [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006, 28(4): 650-656.
- [9] Rhemann C, Hosni A, Bleyer M, *et al.* Fast cost-volume filtering for visual correspondence and beyond [C] // IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), 2011: 3017-3024.
- [10] He K M, Sun J, Tang X O. Guided image filtering [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, 35(6): 1397-1409.
- [11] Scharstein D, Szeliski R. High-accuracy stereo depth maps using structured light [C] // 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003: 195-202.
- [12] Kolmogorov V, Zabih R. Computing visual correspondence with occlusions using graph cuts[C] // Proceedings Eighth IEEE International Conference on Computer Vision (ICCV 2001), 2001: 508-515.
- [13] Humenberger M, Zinner C, Weber M, *et al.* A fast stereo matching algorithm suitable for embedded real-time systems [J]. *Computer Vision and Image Understanding*, 2010, 114(11): 1180-1202.
- [14] Tombari F, Mattoccia S, di Stefano L, *et al.* Near real-time stereo based on effective cost aggregation [C] // 2008 19th International Conference on Pattern Recognition, 2008: 1-4.
- [15] Hirschmuller H. Accurate and efficient stereo processing by semi-global matching and mutual information [C] // 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 2005: 807-814.
- [16] Kowalczyk J, Psota E T, Perez L C. Real-time stereo matching on CUDA using an iterative refinement method for adaptive support-weight correspondences[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2013, 23(1): 94-104.
- [17] Scharstein D, Pal C. Learning conditional random fields for stereo [C] // 2007 IEEE Conference on Computer Vision and Pattern Recognition, 2007: 1-8.
- [18] Yang Q X. A non-local cost aggregation method for stereo matching [C] // 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012: 1402-1409.