

基于水平树结构的可变权重代价聚合立体匹配算法

彭建建, 白瑞林

江南大学轻工过程先进控制教育部重点实验室, 江苏 无锡 214122

摘要 针对基于树结构的代价聚合方法中只利用颜色信息选择权值支持区域时,在图像边界区域易产生误匹配的问题,提出了一种基于水平树结构的可变权重代价聚合立体匹配算法。采用水平树代价聚合得到初始视差值,结合初始视差值与颜色信息重构水平树,在更新后的树结构上进行代价聚合,得到最终视差图。在视差后处理阶段,提出了一种改进的非局部视差后处理算法,将不满足左右一致性匹配的像素点引入匹配代价量构造中,提高了最终视差图的匹配精度。在 Middlebury 数据集的 31 对图像上进行测试,结果表明,未进行视差后处理时所提算法在未遮挡区域的平均误匹配率为 6.96%,代价聚合平均耗时 1.52 s。

关键词 机器视觉; 立体匹配; 代价聚合; 水平树; 非局部视差后处理

中图分类号 TN911.73

文献标识码 A

doi: 10.3788/AOS201838.0115002

Variable Weight Cost Aggregation Algorithm for Stereo Matching Based on Horizontal Tree Structure

Peng Jianjian, Bai Ruilin

Key Laboratory of Advanced Process Control for Light Industry, Ministry of Education, Jiangnan University, Wuxi, Jiangsu 214122, China

Abstract In the cost aggregation methods based on tree structure, the weight support region is selected only by color information, and therefore it is easy to produce mismatching problem in the image boundary area. A variable weight cost aggregation algorithm for stereo matching based on horizontal tree structure is proposed to solve the problem. The initial disparity value is obtained by the cost aggregation of horizontal tree, the horizontal tree is reconstructed with initial disparity value and color information, and the final disparity map is obtained on the updated tree structure by cost aggregation. In the disparity refinement stage, an improved non-local disparity refinement algorithm is proposed with the pixel points that do not satisfy left-right consistency constraint introduced into the matching cost volume, which improves the matching accuracy of final disparity map. Performance evaluation experiments on all 31 Middlebury stereo pairs demonstrate that the proposed algorithm achieves an average error matching rate of 6.96% in the non-occluded areas without disparity refinement, and the cost aggregation takes 1.52 s on average.

Key words machine vision; stereo matching; cost aggregation; horizontal tree; non-local disparity refinement

OCIS codes 150.1135; 100.6890; 330.1400

1 引 言

立体匹配是根据同一场景内不同视角下的两幅或多幅图像,通过寻找像素之间的对应关系来获得物体三维深度信息的过程,是计算机视觉领域中存在已久且尚未解决的问题之一^[1]。Scharstein 等^[2]

将立体匹配算法分为全局算法和局部算法两类。现行的全局匹配算法通常考虑局部的颜色与结构信息,并为图像建立全局能量代价函数,通过置信传播^[3]、图割法^[4]、动态规划^[5]等优化方法为每个像素分配视差值。局部算法利用基于窗口或树结构的代价聚合计算得到像素的视差值,如自适应窗

收稿日期: 2017-06-05; 收到修改稿日期: 2017-08-03

基金项目: 江苏高校优势学科建设工程资助项目(PAPD)、江苏省产学研前瞻性联合研究项目(BY2015019-38)

作者简介: 彭建建(1993—),男,硕士研究生,主要从事机器视觉理论与应用方面的研究。E-mail: peng_wuxi@163.com

导师简介: 白瑞林(1955—),男,教授,博士生导师,主要从事机器视觉与机器人技术方面的研究。

E-mail: bairuilin@hotmail.com(通信联系人)

口^[6]。虽然其精度不如全局算法,但计算复杂度低,效率高,易于实现。局部算法主要包含匹配代价计算、代价聚合、计算视差值和视差后处理 4 个步骤。

代价聚合是局部算法中最重要的一个步骤,其隐含的基本假设为图像局部块具有视差平滑性,因此可以将其看成是在匹配代价基础上进行滤波处理。最简单快速的局部滤波器为盒滤波,但是图像边界会出现视差模糊,即前景膨胀。之后,具有边界保留效果的双边滤波^[7]和引导滤波^[8]被引入代价聚合,可得到与全局方法相媲美的视差图,然而局部方法需要预先定义支持窗口的大小,大的支持窗口会使边界区域过于平滑,而小的支持窗口会导致纹理区域的误匹配。为了解决聚合阶段受预先定义的支持窗口大小限制的问题,相继提出了基于递归^[9]、水平树^[10-11]、最小生成树^[12]、分割树^[13-14]的非局部方法,对图像中任一像素选择的支持窗口为整幅图像,有效避免了在无纹理区域支持窗口过小的情况。与局部方法相比,非局部方法在运算速度和无纹理区域的视差精度方面都有较大的改善。然而,当前大多数非局部方法只利用颜色信息判断相邻像素是否属于相同的视差值,未考虑实际场景中颜色相同而视差不同或颜色不同而视差相同的情况,从而导致图像背景区域和相同颜色边界区域视差估计不准确。

视差后处理是立体匹配算法中的最后一个步骤,其视差精度决定着立体匹配的最终精度。加权中值滤波^[15-16]是应用非常广泛的一种视差后处理方法,通过对相邻区域的视差值进行直方图加权,选择中值对应的视差值作为当前像素的视差,能有效抑制错误的视差值。Yang^[12]基于最小生成树提出了一种非局部视差后处理方法,通过左右一致性匹配将所有像素分成稳定点和非稳定点,基于稳定点重新构造匹配代价量,再次代价聚合得到最终视差值。与加权中值滤波相比,该方法计算速度快,视差精度高,但是忽视了不满足左右一致性匹配而视差估计准确的像素点,因此在新匹配代价量的计算中没有发挥任何作用。

针对图像中颜色边界未必是视差边界的问题,本文提出了一种基于水平树结构的可变权重代价聚合立体匹配算法。该算法对传统的非局部方法只利用颜色信息计算相邻像素的权重进行了改进,引入代价聚合后的初始视差值,通过再次代价聚合使算法在边界区域取得很好的匹配结果,与未引入初始视差值得到的结果相比,精度有所提高。在视差后处理步骤中,提出了一种改进的匹配代价量构造方

式,在未增加额外计算量的条件下,提高了视差精度。

2 算法描述

2.1 匹配代价计算

匹配代价用来衡量同一场景中不同视角下的两幅或多幅图像,在不同视差下对应像素点之间的相似关系。匹配代价计算用 $f: R^{W \times H \times 3} \times R^{W \times H \times 3} \rightarrow R^{W \times H \times L}$ 表示,其中 W, H 表示图像分辨率的宽和高,上标 3 表示图像的 R、G、B 3 个通道, L 表示两对对应像素点之间的最大视差值 d_{\max} 。不同视角下的左右图像 I_L, I_R 在视差为 $0 \sim d_{\max}$ 时的匹配代价量可表示为

$$C = f(I_L, I_R). \quad (1)$$

对于图像中每一个像素点 (x, y) , 在视差 d 下的匹配代价为 $C(x, y, d)$, 将图像的亮度信息与水平梯度信息相结合计算匹配代价:

$$C(x, y, d) = \alpha \cdot \min \left[\frac{1}{3} \sum_{k \in \{R, G, B\}} |I_L^k(x, y + d) - I_R^k(x, y)|, T_c \right] + (1 - \alpha) \cdot \min [|\nabla_x I_L(x, y + d) - \nabla_x I_R(x, y)|, T_g], \quad (2)$$

式中 k 表示图像的 R、G、B 3 个通道; $\nabla_x I_L, \nabla_x I_R$ 分别表示左右彩色图像转换为灰度图像后的水平方向梯度; T_c, T_g 为截断阈值, $T_c = 7, T_g = 2$; α 为平衡因子,取值为 0.11。

利用(2)式计算 I_L 中每一个像素点在视差 $0 \sim d_{\max}$ 下的匹配代价,得到一个维度为 $W \times H \times L$ 的三维矩阵,即最终的匹配代价量。

2.2 代价聚合

由于单个像素的匹配代价区分度不强,且易受噪声影响,基于局部视差平滑性假设,利用相邻像素的信息进行代价聚合,以提高视差区分度。代价聚合可表示为

$$C'_i = \frac{1}{N_i} \sum_{j \in \Omega} W_{i,j}(I) C_j, \quad (3)$$

式中 $N_i = \sum_{j \in \Omega} W_{i,j}(I)$ 为归一化常数, $W_{i,j}(I)$ 表示像素 j 对像素 i 的支持权重, I 为引导图像(一般通过对左右图像盒滤波获得), Ω 表示支持区域范围; C' 、 C 分别为聚合后和聚合前的匹配代价量。

计算每对像素之间的支持权重,采用基于水平树结构的权值传播代价聚合算法^[10],首先基于引导图像 I 构建四连通权值树,如图 1 所示。每个像素被当成一个节点,像素 i, j 之间的支持权重定义为

$$W_{i,j}(I) = \prod_{p,q \in p_{i,j}} T_{p,q}(I), \quad (4)$$

式中 $p, q \in p_{i,j}$ 表示在像素 i, j 之间权值传播路径上的两相邻节点, $T_{p,q}(I)$ 为相邻两节点之间的支持权重, 定义为

$$T_{p,q}(I) = \exp\left[-\frac{\max(|I_p^R - I_q^R|, |I_p^G - I_q^G|, |I_p^B - I_q^B|)}{\sigma}\right], \quad (5)$$

式中 R、G、B 为像素 3 个通道; σ 表示平滑参数, 用于调节相邻像素之间的支持权重。

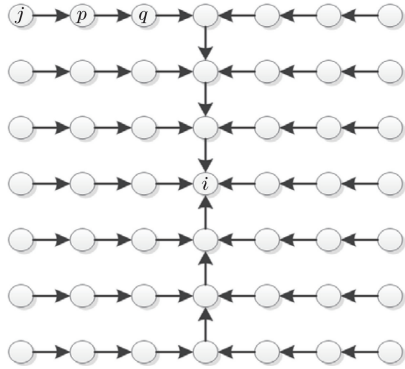


图 1 基于水平树结构的权值传播

Fig. 1 Weight propagation based on horizontal tree structure

绝大多数代价聚合方法只是在单独视差下对匹配代价进行滤波, 未考虑倾斜平面不满足图像局部区域视差相近原则的情况, 因此采用文献[11]提出的方法, 将正则项引入匹配代价聚合, 并基于上述建立的树结构, 匹配代价沿着图 1 路径聚合, 最终匹配代价量为

$$C_p(d_p) = m_p(d_p) + \sum_{q \in \nu(p)} \min_{d_q \in D} [s(d_p, d_q) + C_q(d_q)] W_{p,q}(I), \quad (6)$$

式中 $m_p(d_p)$ 为未进行代价聚合前像素 p 在视差 d_p 下的匹配代价; $\nu(p)$ 表示以 p 为根节点的所有叶节点, 亦为图像中所有节点; D 为视差搜索范围, $W_{p,q}(I)$ 为(4)式定义的像素 p, q 之间的支持权重; $s(d_p, d_q)$ 表示平滑惩戒因子, 定义为

$$s(d_p, d_q) = \begin{cases} 0, & |d_p - d_q| = 0 \\ p_{\text{smooth}}, & 0 < |d_p - d_q| \leq 1, \\ \infty, & \text{else} \end{cases} \quad (7)$$

式中 p_{smooth} 为常量, 取值为 2。最后采用 Winner-Takes-All 策略, 在视差区间内选择匹配代价最小的点作为最优点对 p 点进行视差选择, 获得初始视差

$$d_0(p) = \arg \min_{d \in D} [C_p(d_p)]. \quad (8)$$

2.3 代价再聚合

只基于颜色信息选择像素权重支持区域对图像边界视差估计不准确, 初始视差图中还存在大量误匹配, 因此引入初始视差重构水平树结构, 从而改善匹配精度。与 2.2 节代价聚合方式不同, 代价再聚合重构水平树中相邻两节点之间的支持权重定义为

$$\omega_{p,q} = (1 - k) \max(|I_p^R - I_q^R|, |I_p^G - I_q^G|, |I_p^B - I_q^B|) + k |d_p - d_q|, \quad (9)$$

$$T_{p,q}(I) = \exp\left(-\frac{\omega_{p,q}}{\sigma}\right), \quad (10)$$

式中 k 为颜色和视差权重平衡因子, $k \in [0, 1]$; d_p, d_q 分别为像素 p, q 的初始视差。利用重构后的水平树和(4)、(6)~(8)式得到代价再聚合后的视差值。图 2 为采用代价再聚合前后中心像素的权值支持区域图。如图 2(a) 所示, 图像中心像素(红色方框)与其邻域像素的视差值相近, 而颜色信息相差很

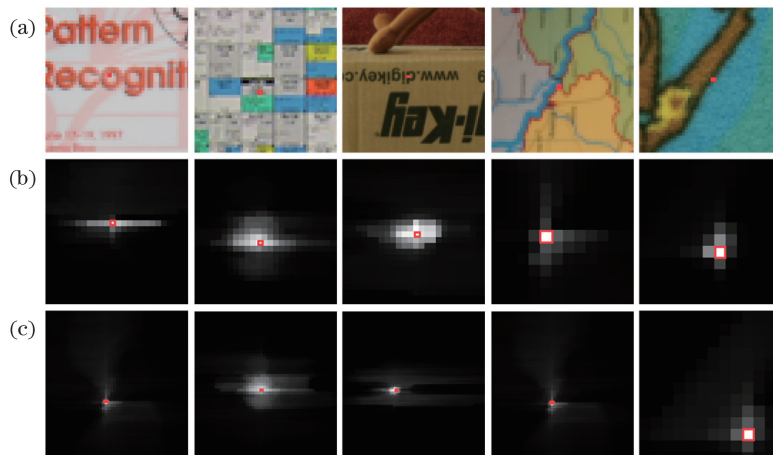


图 2 权值支持区域图(红色方框为中心像素, 邻域像素亮度值表示对中心像素的支持权重)。(a) 图像局部块;

(b) 未采用代价再聚合的支持权重图; (c) 采用代价再聚合的支持权重图

Fig. 2 Support weights for selected regions (the brightness value of neighborhood pixel represents the support weight for central pixel which is marked with red box). (a) Image partial block; (b) support weight map without iterative cost aggregation; (c) support weight map after iterative cost aggregation

大,只利用颜色信息计算支持权重得到如图 2(b)所示的支持权重图。因支持区域过小易产生误匹配,图 2(c)为引入初始视差获得的支持权重图,支持区域明显扩大,有效降低了像素匹配歧义性。

如图 3(c)所示,未采用代价再聚合时,在视差

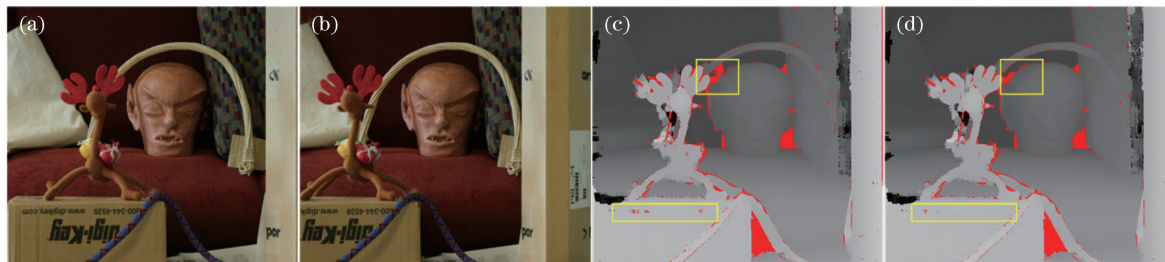


图 3 采用代价再聚合前后的视差变化图。(a) Reindeer 左图像;(b) Reindeer 右图像;
(c)未采用代价再聚合得到的左视差图;(d)采用代价再聚合得到的左视差图

Fig. 3 Disparity variation maps computed before and after iterative cost aggregation. (a) Reindeer left image;
(b) Reindeer right image; (c) left disparity map computed before iterative cost aggregation;
(d) left disparity map computed after iterative cost aggregation

2.4 视差后处理

通过上述方法得到的初始视差在遮挡区域(左右图像不能同时观测区域)存在大量误匹配,需要进行后处理。首先通过左右一致性来检测匹配异常点,判定公式为

$$d_L(x,y) = d_R[x - \max(d_L, 0), y], \quad (11)$$

式中 d_L, d_R 为左右初始视差图。不满足(11)式的点即可视为异常点,否则为稳定点。利用稳定点与异常点重新构造匹配代价量,构建方式定义为

$$C_d^{new}(p) = \begin{cases} |d - D(p)|, & p \text{ is a stable point and } D(p) > 0 \\ k_1 |d - D(p)|, & p \text{ is an unstable point} \end{cases}, \quad (12)$$

式中 $D(p)$ 表示初始左视差图; $d \in [0, d_{max}]$; k_1 为比例因子,用于调节异常点对匹配代价量的贡献,取

不连续和图像边界区域均存在大量的误匹配点(红色区域),得到的视差图比较粗糙。图 3(d)为采用代价再聚合后得到的视差图,与图 3(c)相比,在图像颜色边界和背景区域视差估计方面有明显改善(黄色方框内),提高了匹配精度。

值范围为 $[0, 1]$ 。最后,利用 2.3 节代价再聚合生成的水平树对新匹配代价量进行代价聚合,采用 Winner-Takes-All 策略获得最终视差图。

图 4(c)为左右一致性检测后得到的视差图,其中黑色区域像素为异常点。图 4(d)为进行视差后处理之前的初始左视差图。文献[11]的视差后处理阶段未考虑黄色方框区域内的准确视差值对新匹配代价量的积极作用,最终视差图在该区域还存在大量误匹配。图 5 为标准 Dolls 图像利用改进后的视差后处理算法获得的视差图。与文献[11]的视差后处理方法相比,两者匹配代价计算、代价聚合处理方法相同。图 5 中红色区域为误匹配像素点,误差阈值为 1。改进方法获得的视差图在不满足左右一致性的检测区域有明显改善,提高了匹配精度。

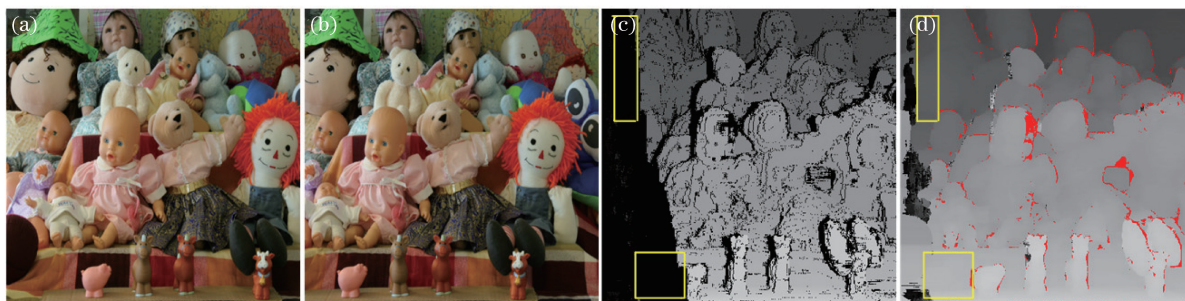


图 4 视差后处理的机理。(a) Dolls 左图像;(b) Dolls 右图像;(c)左右一致性检测;
(d)未进行视差后处理的初始左视差图

Fig. 4 Principle of disparity refinement. (a) Dolls left image; (b) Dolls right image; (c) left-right consistence test;
(d) initial left disparity map without disparity refinement

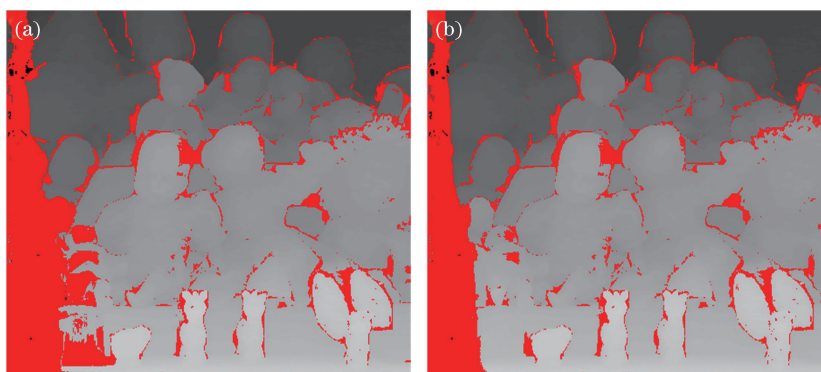


图 5 采用不同视差后处理方法得到的视差图。(a)文献[11]的视差后处理方法;(b)改进的视差后处理方法
Fig. 5 Disparity maps with different disparity refinement methods. (a) Disparity refinement method in Ref. [11];
(b) improved disparity refinement method

3 实验结果与分析

为了验证算法的有效性,选用 Middlebury 测试平台提供的图像进行测试。计算机配置为 Pentium E6700 3.20 GHz 主频 CPU、2G 内存。实验中算法涉及的具体参数如表 1 所示。

表 1 本文立体匹配算法参数

Table 1 Parameters of the proposed stereo matching algorithm

Parameter	α	T_c	T_g	σ	p_{smooth}	k	k_1
Value	0.11	7	2	255×0.08	2	0.5	0.1

3.1 代价聚合算法性能测试

为了验证本文提出的代价聚合算法的性能,将

所提算法与近年来的前沿局部立体匹配算法进行比较,包括最小生成树(MST)^[12]、引导滤波(GF)^[8]、跨尺度立体匹配^[17][其中包含跨尺度最小生成树(CS-MST)、跨尺度分割树(CS-ST)]以及基于水平树平滑正则化的代价聚合立体匹配算法(LSECVR)^[11]。评判标准为代价聚合后的初始视差图在非遮挡区域的匹配精度,其中设置误差阈值为 1。为了更好地展示对比实验的效果,将误匹配像素点用红色标记,实验结果如图 6 所示,可以看出红色区域明显减少。为验证改进算法的普遍适用性,在 Middlebury 数据集的 31 对图像上进行测试,非遮挡区域的匹配误差如表 2 所示,其中下标的数字表示各算法匹配精度

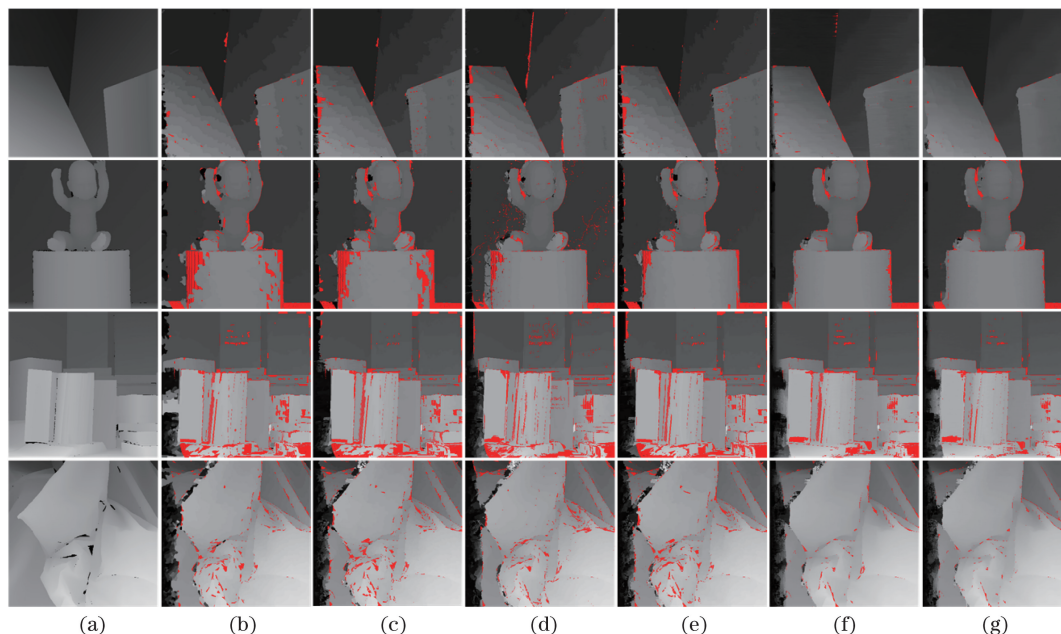


图 6 不同代价聚合算法得到的视差图(红色区域为误匹配像素点)。(a)真实视差图;(b)最小生成树;(c)跨尺度最小生成树;(d)引导滤波;(e)跨尺度分割树;(f)文献[11]代价聚合算法;(g)改进的代价聚合算法

Fig. 6 Disparity maps obtained by different cost aggregation algorithms (mismatched pixels are marked in red area).

(a) Real disparity map; (b) by minimum spanning tree; (c) by cross-scale minimum spanning tree; (d) by guided filtering; (e) by cross-scale segment tree; (f) by cost aggregation algorithm in Ref. [11]; (g) by improved cost aggregation algorithm

表 2 不同立体匹配算法在未进行视差后处理的非遮挡区域中的匹配误差(单位:%)

Table 2 Error of different stereo matching methods in non-occluded areas without disparity refinement (unit: %)

Stereo pair	MST	CS-MST	GF	CS-ST	LSECVR	Proposed
Tsukuba	2.12 ₄	1.57 ₁	2.51 ₆	1.74 ₂	2.29 ₅	1.77 ₃
Venus	0.84 ₃	1.38 ₄	2.03 ₆	1.45 ₅	0.56 ₂	0.34 ₁
Teddy	7.61 ₅	5.53 ₃	8.48 ₆	6.07 ₄	4.91 ₂	4.25 ₁
Cones	4.10 ₄	4.15 ₅	3.61 ₃	4.42 ₆	3.44 ₂	3.36 ₁
Aloe	4.14 ₃	4.63 ₄	5.53 ₆	4.71 ₅	2.88 ₂	2.67 ₁
Art	9.79 ₄	10.79 ₆	9.03 ₃	10.50 ₅	6.72 ₂	6.46 ₁
Baby1	7.37 ₅	8.39 ₆	4.69 ₄	4.53 ₃	2.88 ₂	2.59 ₁
Baby2	11.95 ₄	13.37 ₅	6.08 ₃	15.11 ₆	2.61 ₂	1.60 ₁
Baby3	5.64 ₃	7.25 ₆	5.79 ₄	6.23 ₅	3.68 ₂	3.66 ₁
Books	9.56 ₃	10.26 ₆	10.22 ₄	10.24 ₅	6.71 ₂	5.63 ₁
Bowling1	16.81 ₄	20.89 ₅	14.52 ₃	21.72 ₆	8.82 ₂	6.59 ₁
Bowling2	9.31 ₄	10.15 ₅	7.08 ₃	11.18 ₆	4.88 ₂	3.40 ₁
Cloth1	0.51 ₃	0.61 ₄	1.08 ₆	0.66 ₅	0.27 ₂	0.15 ₁
Cloth2	2.85 ₃	4.13 ₆	3.46 ₄	4.04 ₅	1.43 ₂	1.07 ₁
Cloth3	1.77 ₃	2.66 ₅	2.15 ₄	2.72 ₆	1.41 ₂	1.06 ₁
Cloth4	1.30 ₃	1.87 ₆	1.62 ₄	1.75 ₅	1.13 ₂	1.10 ₁
Dolls	5.00 ₃	5.95 ₆	5.04 ₄	5.52 ₅	3.11 ₂	2.90 ₁
Flowerpots	16.67 ₅	19.41 ₆	12.79 ₃	15.22 ₄	12.66 ₂	11.43 ₁
Lampshade1	10.43 ₃	11.99 ₆	11.57 ₅	10.61 ₄	9.00 ₂	8.22 ₁
Lampshade2	20.88 ₅	18.20 ₄	21.13 ₆	12.08 ₃	7.42 ₂	5.78 ₁
Laundry	13.69 ₄	12.94 ₃	16.40 ₆	14.51 ₅	11.07 ₂	10.70 ₁
Midd1	32.32 ₅	27.85 ₃	40.11 ₆	26.95 ₁	27.62 ₂	29.52 ₄
Midd2	34.50 ₅	32.09 ₄	35.85 ₆	24.56 ₁	25.51 ₃	25.09 ₂
Moebius	7.67 ₁	8.69 ₅	9.25 ₆	8.55 ₄	8.11 ₂	8.16 ₃
Monopoly	22.51 ₁	24.21 ₂	27.99 ₆	25.50 ₃	26.37 ₄	27.14 ₅
Plastic	42.53 ₄	47.03 ₆	39.29 ₂	42.72 ₅	40.71 ₃	34.87 ₁
Reindeer	9.15 ₅	9.87 ₆	7.23 ₃	8.33 ₄	5.08 ₂	3.67 ₁
Rocks1	2.23 ₃	2.83 ₆	2.70 ₅	2.64 ₄	1.14 ₂	0.91 ₁
Rocks2	1.57 ₃	2.08 ₆	1.61 ₄	1.90 ₅	0.81 ₂	0.78 ₁
Wood1	8.68 ₅	11.06 ₆	4.83 ₃	5.96 ₄	0.24 ₁	0.25 ₂
Wood2	0.99 ₃	5.61 ₅	2.34 ₄	6.42 ₆	0.63 ₂	0.62 ₁
Average rank	3.65 ₃	4.87 ₆	4.45 ₅	4.42 ₄	2.19 ₂	1.42 ₁
Average error	10.56	11.21	10.52	10.28	7.55	6.96

的排列顺序。从表 2 可以看出,改进算法的平均排名和平均匹配精度都优于其他几种算法,平均匹配误差仅为 6.96%,相比于引导滤波算法和最小生成树,平均匹配误差分别降低了 3.56%和 3.6%。此外,代价聚合平均运行时间为 1.52 s,近似满足实时性要求。

3.2 视差后处理算法性能测试

为了验证本文提出的视差后处理算法的性能,将所提视差后处理算法与文献[11]的后处理方法进行

比较,其中匹配代价计算和代价聚合与本文采用方法相同,避免了匹配代价计算和代价聚合方式的不同对实验结果的影响。在 Middlebury 数据集的 31 对图像上进行测试,实验结果如表 3 所示,其中下标数字为匹配精度排名,评价标准为全图像区域匹配误差百分比,误差阈值设置为 1。从表 3 可以看出,与文献[11]的方法相比,本文提出的视差后处理算法的全图像平均匹配误差下降了 1.93%。此外,两种视差后处理方

法(不包括匹配代价计算和代价聚合)耗时分别为 0.826 s(文献[11]方法)和 0.839 s(本文方法),几乎相等,本文算法在未增加计算量的情况下提高了匹配精度。图 7 直观展示了本文视差后处理算法相对于文

献[11]后处理方法的优点,其中红色区域为误匹配像素。从图中可以看出,改进的视差后处理方法在不满足左右一致性匹配区域的匹配精度有所提高,但是视差图中仍存在一些误匹配点,需进一步改进。

表 3 不同立体匹配算法视差后处理在全图像区域的匹配误差(单位:%)

Table 3 Matching error of different stereo matching methods in image areas with disparity refinement (unit: %)

Stereo pair	LSECVR	Proposed	Stereo pair	LSECVR	Proposed
Tsukuba	3.63 ₁	4.01 ₂	Dolls	17.81 ₂	12.81 ₁
Venus	2.48 ₁	3.23 ₂	Flowerpots	20.83 ₁	22.26 ₂
Teddy	16.08 ₂	11.83 ₁	Lampshade1	23.84 ₂	22.29 ₁
Cones	14.11 ₂	11.26 ₁	Lampshade2	28.79 ₂	21.03 ₁
Aloe	11.37 ₁	11.86 ₂	Laundry	26.56 ₂	23.26 ₁
Art	25.41 ₂	21.37 ₁	Midd1	38.73 ₂	34.27 ₁
Baby1	8.80 ₂	8.45 ₁	Midd2	33.14 ₂	28.71 ₁
Baby2	9.79 ₂	5.78 ₁	Moebius	18.66 ₁	19.08 ₂
Baby3	14.48 ₁	15.61 ₂	Monopoly	34.03 ₂	33.65 ₁
Books	18.93 ₂	18.01 ₁	Plastic	41.28 ₁	41.67 ₂
Bowling1	26.19 ₂	23.44 ₁	Reindeer	15.93 ₂	13.92 ₁
Bowling2	19.14 ₂	18.90 ₁	Rocks1	13.13 ₂	10.06 ₁
Cloth1	14.55 ₂	10.34 ₁	Rocks2	14.55 ₂	10.35 ₁
Cloth2	17.03 ₂	12.46 ₁	Wood1	9.16 ₁	9.22 ₂
Cloth3	11.62 ₂	11.41 ₁	Wood2	8.80 ₁	8.82 ₂
Cloth4	18.17 ₂	17.73 ₁	Average rank	1.71 ₂	1.29 ₁
Average error	18.61	16.68			

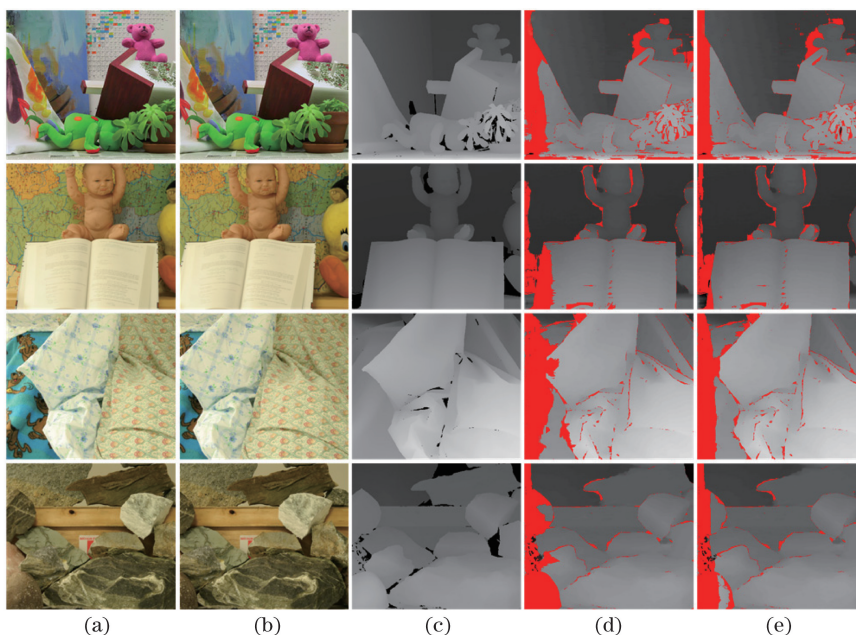


图 7 Middlebury 标准图像实验结果。(a)左参考图像;(b)右参考图像;(c)左真实视差图;
(d)文献[11]的视差后处理方法;(e)改进的视差后处理方法

Fig. 7 Experimental results of the Middlebury benchmark images. (a) Left reference images; (b) right reference images; (c) left real disparity maps; (d) by disparity refinement method in Ref. [11]; (e) by improved disparity refinement method

4 结 论

针对初始匹配视差在图像颜色边界存在的误匹配问题,提出了一种基于水平树结构的可变权重代价聚合立体匹配算法。将初始代价聚合获得的视差图引入到水平权值树构造中,有效缓解了图像在颜色相同但深度不同以及颜色不同但深度相同区域的误匹配问题。在视差后处理阶段,引入视差对匹配代价的权重系数,有效降低了左右视差不一致区域的误匹配率。本文算法在非遮挡区域的匹配精度高于其他基于滤波器的方法,而在遮挡区域的匹配精度与某些全局算法有一定差距。在今后的研究中,将重点关注提高遮挡区域的匹配精度。

参 考 文 献

- [1] Bleyer M, Breiteneder C. Stereo matching—state-of-the-art and research challenges[M]. Advanced topics in computer vision. London: Springer, 2013: 143-179.
- [2] Scharstein D, Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms[J]. International Journal of Computer Vision, 2002, 47: 7-42.
- [3] Sun J, Zheng N N, Shum H Y. Stereo matching using belief propagation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(7): 787-800.
- [4] Boykov Y, Veksler O, Zabih R. Fast approximate energy minimization via graph cuts[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(11): 1222-1239.
- [5] Bleyer M, Gelautz M. Simple but effective tree structures for dynamic programming-based stereo matching[C]//International Conference on Computer Vision Theory and Applications, Funchal, Portugal, 2008: 415-422.
- [6] Zhu S P, Li Z. A stereo matching algorithm using improved gradient and adaptive window[J]. Acta Optica Sinica, 2015, 35(1): 0110003.
祝世平, 李政. 基于改进梯度和自适应窗口的立体匹配算法[J]. 光学学报, 2015, 35(1): 0110003.
- [7] 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.
- [8] Hosni A, Rhemann C, Bleyer M, *et al.* Fast cost-volume filtering for visual correspondence and beyond[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(2): 504-511.
- [9] Cigla C. Recursive edge-aware filters for stereo matching[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015: 27-34.
- [10] Yang Q, Li D, Wang L, *et al.* Full-image guided filtering for fast stereo matching [J]. IEEE Signal Processing Letters, 2013, 20(3): 237-240.
- [11] Yang Q. Local smoothness enforced cost volume regularization for fast stereo correspondence[J]. IEEE Signal Processing Letters, 2015, 22(9): 1429-1433.
- [12] Yang Q. A non-local cost aggregation method for stereo matching[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2012: 1402-1409.
- [13] Mei X, Sun X, Dong W M, *et al.* Segment-tree based cost aggregation for stereo matching[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013: 313-320.
- [14] Yao P, Zhang H, Xue Y, *et al.* Segment-tree based cost aggregation for stereo matching with enhanced segmentation advantage[C]//IEEE International Conference on Acoustics, Speech and Signal Processing, 2017: 2027-2031.
- [15] Ma Z Y, He K M, Wei Y C, *et al.* Constant time weighted median filtering for stereo matching and beyond[C]//Proceedings of the IEEE International Conference on Computer Vision, 2013: 49-56.
- [16] Sun X, Mei X, Jiao S H, *et al.* Stereo matching with reliable disparity propagation[C]//IEEE International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, 2011: 132-139.
- [17] Zhang K, Fang Y, Min D, *et al.* Cross-scale cost aggregation for stereo matching[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014: 1590-1597.