

## 改进的代价聚集快速立体匹配算法

杨 罡<sup>1</sup>, 晋 涛<sup>1</sup>, 王大伟<sup>1</sup>, 曹京津<sup>1</sup>, 张 娜<sup>1</sup>, 严碧武<sup>2</sup>, 李 涛<sup>2</sup>, 程 远<sup>2</sup>

(1. 国网山西省电力科学研究院, 太原 030045;

2. 国网电力科学研究院武汉南瑞有限责任公司, 武汉 430074)

**摘 要:** 针对立体匹配中的代价聚集问题, 提出一种改进的代价聚集算法。对图像进行超像素分割并建立最小生成树, 采用树形滤波器进行代价聚集, 生成超像素代价。利用权重融合初始像素代价和超像素代价, 得到最终的像素代价。实验结果表明, 与最小生成树、线段树等算法相比, 该算法的时间复杂度较低, 且生成的视差图具有良好的边缘保持特性。

**关键词:** 代价聚集; 超像素; 最小生成树; 树形滤波器; 立体匹配

**中文引用格式:** 杨罡, 晋涛, 王大伟, 等. 改进的代价聚集快速立体匹配算法[J]. 计算机工程, 2019, 45(7): 268-272, 281.

**英文引用格式:** YANG Gang, JIN Tao, WANG Dawei, et al. Improved cost aggregation algorithm for fast stereo matching[J]. Computer Engineering, 2019, 45(7): 268-272, 281.

## Improved Cost Aggregation Algorithm for Fast Stereo Matching

YANG Gang<sup>1</sup>, JIN Tao<sup>1</sup>, WANG Dawei<sup>1</sup>, CAO Jingjin<sup>1</sup>, ZHANG Na<sup>1</sup>, YAN Biwu<sup>2</sup>, LI Tao<sup>2</sup>, CHENG Yuan<sup>2</sup>

(1. State Grid Shanxi Electric Power Research Institute, Taiyuan 030045, China;

2. Wuhan Nanrui Limited Company of State Grid Electric Power Research Institute, Wuhan 430074, China)

**[Abstract]** For the cost aggregation problem in stereo matching, an improved cost aggregation algorithm is proposed. The image is segmented by superpixel and the Minimum Spanning Tree (MST) is established. Then, use tree filter for cost aggregation to generate superpixel cost. The final pixel cost is obtained by weighting the initial pixel cost and superpixel cost. Experimental results show that compared with MST, Segmcat-Tree (ST) and other methods, the proposed algorithm has lower time complexity, and can get disparity map with good edge preserve features.

**[Key words]** cost aggregation; superpixel; Minimum Spanning Tree (MST); tree filter; stereo matching

**DOI:** 10.19678/j.issn.1000-3428.0050790

### 0 概述

在计算机视觉领域, 立体匹配技术一直是研究的热点和难点。现有的立体匹配方法主要分为局部匹配和全局匹配 2 类。全局立体匹配方法通过建立能量方程并对其进行最小化来获取视差, 采用的技术主要有置信度传播<sup>[1]</sup>、动态规划<sup>[2-3]</sup>、图割<sup>[4-6]</sup>等。虽然全局匹配方法能获得很好的视差, 但是其复杂度较高, 应用性不强。局部立体匹配方法只考虑像素邻域范围内的像素点, 与全局立体匹配方法相比复杂度较低。文献[7]把局部匹配方法分为代价计算、代价聚集、视差求取、视差求精 4 个步骤, 其中, 代价聚集被认为是局部立体匹配方法的核心。

文献[8]利用  $O(1)$  复杂度的双边滤波器进行代价聚合, 因其受制于快速双边滤波时的量化手段,

算法的运行结果并不好。文献[9]提出将原始的彩色图作为导向图来进行导向滤波代价聚集, 由于其假设视差数据在结构上和导向图相似, 因此该方法的效果比双边滤波更好。文献[10]依据像素相似度, 对每个像素建立支撑区域, 其代价聚集取得了良好的效果。基于窗口或支撑区域的方法, 其待聚集区域在本质上是局部的, 文献[11-12]提出非局部代价聚集的思想, 将图像像素点看作图的节点, 像素点的颜色差异作为边的权重, 来建立最小生成树。该方法考虑了图像中全局像素对于待聚集像素的影响, 因此取得了较好的效果。文献[13]将卷积神经网络应用于代价聚集, 而文献[14]采用多棵最小分离树来进行 3D 代价聚集, 虽然两者都取得了良好的效果, 但其依赖于数据驱动或强大的硬件设备, 时间成本较高。

**基金项目:** 国网山西电力 2017 年重点研究项目 (52053016000v)。

**作者简介:** 杨 罡 (1983—), 男, 高级工程师, 主研方向为图像处理、高压线路智能监测研究; 晋 涛, 高级工程师; 王大伟、曹京津、张 娜, 工程师; 严碧武, 高级工程师; 李 涛, 工程师; 程 远, 助理工程师。

**收稿日期:** 2018-03-15 **修回日期:** 2018-05-22 **E-mail:** yanggang9800@163.com

文献[15]以超像素作为节点进行随机游走, 本文将其与文献[11-12]采用的非局部代价聚集方法相结合, 提出一种改进的代价聚集算法。通过简单线性迭代聚类 (Simple Linear Iterative Cluster, SLIC) 算法进行超像素分割, 构建超像素最小生成树, 计算超像素代价。以一定的权重融合初始像素代价和超像素代价, 采用 winner take all 进行视差后处理, 得到立体匹配结果。

## 1 基于超像素的快速代价聚集

本文提出的代价聚集方法流程如图 1 所示。

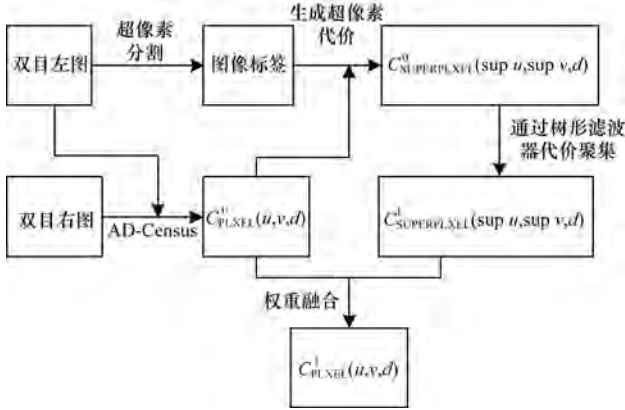


图 1 基于超像素的快速代价聚集流程

应用 SLIC 算法生成图像的超像素。利用克鲁斯卡尔算法生成超像素最小生成树, 通过树形滤波器生成超像素代价。基于权重融合 AD-Census 代价匹配得到的像素代价  $C_{\text{PIXEL}}^0(u, v, d)$  和超像素代价  $C_{\text{SUPERPIXEL}}^1(\sup u, \sup v, d)$ , 生成像素代价  $C_{\text{PIXEL}}^1(u, v, d)$ , 应用 winner take all 策略得到视差数据。

### 1.1 AD-Census 代价匹配

采用 AD-Census 代价计算的方法生成初始像素代价  $C_{\text{PIXEL}}^0(u, v, d)$ 。AD-Census 代价由 2 个部分构成: 绝对误差 (Absolute Difference, AD) 变换和 Census 变换。

AD 变换通过计算双目图像中 2 个像素的颜色差异得到匹配代价, 计算过程如下:

$$C_{\text{AD}}(u, v, d) = \frac{1}{3} \sum_{i=R, G, B} |I_i^{\text{Left}}(u, v) - I_i^{\text{Right}}(i - d, v)| \quad (1)$$

Census 变换是一种非参数变换。首先对待匹配双目图像进行 Census 变换, 然后比较双目图像中 2 个像素的 Census 编码, 得到汉明距离并将其作为匹配代价。本文采用  $5 \times 5$  的窗口对图像进行 Census 变换。任一像素点  $(u, v)$  的 Census 变换通过衡量它与周边像素的亮度差异得到, 过程如下:

$$\text{Census}(u, v) = \bigotimes_{(m, n) \in N(u, v)} \varepsilon(I(m, n), I(u, v)) \quad (2)$$

$$\varepsilon(a, b) = \begin{cases} 1, & b > a \\ 0, & b \leq a \end{cases} \quad (3)$$

通过异或操作比较双目图像中 2 个像素的二进制串, 从而生成汉明距离。对于  $5 \times 5$  的窗口, 汉明距离为 0 代表 2 个像素的相似度最高, 汉明距离为 25 代表其相似度最低。由此可以得到 Census 变换的代价计算公式:

$$C_{\text{CENSUS}}(u, v, d) = \text{Ham}(\text{Census}^{\text{Left}}(u, v), \text{Census}^{\text{Right}}(u - d, v)) \quad (4)$$

文献[13]提到 Census 变换对于噪声和光照变换的鲁棒性高, 而对于重复纹理区域以及弱纹理区域的匹配效果较差。而 AD 变换的特性恰好能够和 Census 变换进行互补, 因此可以通过级联 AD 与 Census 组成 AD-Census 算法, 从而获得鲁棒性更高的初始像素代价, 具体如下:

$$C_{\text{PIXEL}}^0(u, v, d) = p(C_{\text{CENSUS}}(u, v, d), \lambda_{\text{CENSUS}}) + p(C_{\text{AD}}(u, v, d), \lambda_{\text{AD}}) \quad (5)$$

其中,  $\lambda_{\text{CENSUS}}$  和  $\lambda_{\text{AD}}$  是融合系数,  $p$  代表代价和融合系数相乘。

### 1.2 超像素最小生成树

通过 AD-Census 方法计算得到的像素代价中包含噪声, 因此需要通过代价聚集步骤来降低匹配的模糊性和代价中的噪声。代价聚集方法基于相似的区域具有相似的视差这一特征, 传统的代价聚集方法都是针对像素代价计算进行的, 其复杂度依赖于像素数目。文献[15]引入超像素分割的思想, 将相似的像素组成超像素, 由于超像素数目远小于像素数目, 因此超像素代价聚集计算比像素代价聚集计算的速度更快。文献[15]通过像素代价构造超像素代价, 并重启随机游走算法对其进行迭代优化。随机游走过程需要进行多次迭代, 并且超像素构成的大型邻接稀疏矩阵每次都要参与运算, 因此, 算法的时间性能和空间复杂度都较差。本文算法基于超像素分割生成超像素最小生成树, 并通过树形滤波器进行代价聚集。在代价聚集的过程中, 只需要进行少量计算就可获得代价聚集结果。

采用超像素进行代价聚集是基于 2 个方面的考虑。首先, 超像素按照相似度对图像进行分割, 超像素代价  $C_{\text{SUPERPIXEL}}^0(\sup u, \sup v, d)$  通过属于同一区域的像素代价  $C_{\text{PIXEL}}^0(u, v, d)$  生成, 因此, 它的抗噪鲁棒性比像素代价  $C_{\text{PIXEL}}^0(u, v, d)$  高。其次, 超像素的数量远小于像素的数量, 其代价聚集的时间大幅缩短, 同时也节约了内存。

本文采用 SLIC 算法<sup>[16]</sup>进行超像素分割, 将一

幅图像分割为若干小块,且小块中像素的相似度很高,超像素实质上代表一块图像区域。图 2 所示为对一幅 450 像素 × 375 像素的图像进行超像素分割后的效果,该图被分割为 200 个超像素。

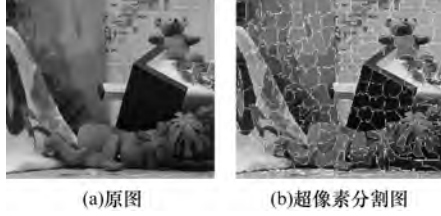


图 2 SLIC 超像素分割效果

由图 2 可以看出,图像中  $450 \times 375 = 168\ 750$  个像素被划分到 200 个超像素中,传统的代价聚集方法需要在 168 750 个像素上进行  $D$  次,现在只需在 200 个超像素上进行  $D$  次。所生成的超像素的颜色、亮度以及超像素代价如下:

$$\begin{cases} Color(\sup u, \sup v) = \frac{\sum_{(u,v) \in (\sup u, \sup v)} Color(u, v)}{\sum_{(u,v) \in (\sup u, \sup v)} 1} \\ Intensity(\sup u, \sup v) = \frac{\sum_{(u,v) \in (\sup u, \sup v)} Intensity(u, v)}{\sum_{(u,v) \in (\sup u, \sup v)} 1} \\ C_{SUPERPIXEL}^0(\sup u, \sup v, d) = \frac{\sum_{(u,v) \in (\sup u, \sup v)} C_{PIXEL}^0(u, v, d)}{\sum_{(u,v) \in (\sup u, \sup v)} 1} \end{cases} \quad (6)$$

为了生成超像素最小生成树,可以把超像素看作是图,将超像素点看作图的一个节点,相邻 2 个超像素构成边,代表 2 个超像素的相邻关系。边上的值为超像素之间的亮度差值,差值越小,超像素的相似度越高。其计算过程如下:

$$\Delta I_{AB} = Intensity(A) - Intensity(B) \quad (7)$$

将亮度差值作为边的权重,构造最小生成树,如图 3 所示。2 个超像素在图中只有一条连通的路径,该路径是连接这 2 个超像素的所有路径中边权重最小的,代表这条路径中的超像素具有最好的相似度并且超像素间不易跨越边缘。利用该路径组成的超像素进行代价聚集,聚集后的视差空间能够较好地保持边缘特性。

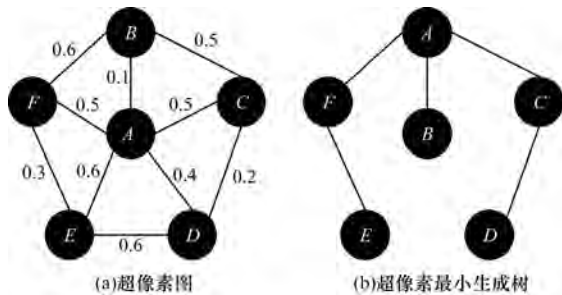


图 3 超像素最小生成树示意图

### 1.3 基于超像素最小生成树的代价聚集

文献[11]将整幅图像中的像素作为图的节点,像素间为 4 邻域或 8 邻域,构建最小生成树。文献[12]在局部区域建立多棵最小生成树。与文献[11-12]相比,本文利用超像素最小生成树建立代价聚集时,由于超像素数目远小于像素数目,因此其速度较快。同时,本文加入超像素作为匹配基础,所以最终生成的像素代价  $C_{PIXEL}^1(u, v, d)$  由超像素代价  $C_{SUPERPIXEL}^1(\sup u, \sup v, d)$  和像素代价  $C_{PIXEL}^0(u, v, d)$  组成。对生成的超像素最小生成树,通过 2 次遍历最小生成树进行代价聚集。初始时需要从最小生成树的叶子节点,依据超像素的孩子节点自底向上更新其代价值,使超像素在代价聚集时受到其底端节点的影响。当第 1 次遍历到达最小生成树的根节点后,进行自顶向下的代价更新,从而增加父节点对子节点代价聚集的影响。图 4 展示了基于超像素最小生成树的代价聚集过程。

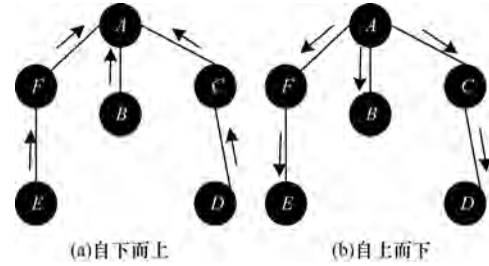


图 4 超像素代价聚集过程

在代价聚集时,将超像素之间的高斯权重作为相邻超像素对当前超像素的贡献权重,计算过程如下:

$$Gauss(A, B) = \exp\left(-\frac{Intensity(A) - Intensity(B)}{2\sigma^2}\right) \quad (8)$$

其中,  $A, B$  代表 2 个超像素,  $\sigma$  代表方差,  $\exp$  代表指数,  $Gauss(A, B)$  为超像素之间的高斯权重。

对于超像素  $A$ ,在其自底向上的代价聚集过程中,超像素的代价更新公式如下:

$$C_{SUPERPIXEL}^1(A, d) = C_{SUPERPIXEL}^0(A, d) + \sum_{Q \in N(A)} Gauss(A, Q) \times C_{SUPERPIXEL}^0(Q, d) \quad (9)$$

其中,  $N(A)$  表示超像素  $A$  的邻域。

在进行自顶向下的代价聚集之后,超像素代价如下:

$$C_{SUPERPIXEL}^1(A, d) = Gauss(Pr(A), A) \times C_{SUPERPIXEL}^1(Pr(A), d) + C_{SUPERPIXEL}^1(A, d) \times (1 - Gauss^2(Pr(A), A)) \quad (10)$$

其中,  $Pr(A)$  代表超像素  $A$  的父节点。

在代价更新的过程中,超像素节点都直接或间接对其他超像素节点的代价更新有影响,由于其计算过

程只需要进行少量的加、减、乘运算, 因此能高效地进行代价聚集。得到超像素代价  $C_{\text{SUPERPIXEL}}^d(\text{sup } u, \text{sup } v, d)$  之后, 利用式(11)将其与初始代价  $C_{\text{PIXEL}}^0(u, v, d)$  进行融合, 最终得到像素代价  $C_{\text{PIXEL}}^d(u, v, d)$ :

$$C_{\text{PIXEL}}^d(u, v, d) = \lambda_{\text{SUPERPIXEL}} C_{\text{SUPERPIXEL}}^d(\text{sup } u, \text{sup } v, d) + \lambda_{\text{PIXEL}} C_{\text{PIXEL}}^0(u, v, d) \quad (11)$$

其中,  $(u, v) \in (\text{sup } u, \text{sup } v)$ 。

利用 winner take all 策略<sup>[7]</sup>对像素代价  $C_{\text{PIXEL}}^d(u, v, d)$  进行计算, 得到视差, 并采用文献[17]的视差后处理方法对视差求精。

## 2 实验结果与分析

为验证本文算法的优越性, 在 Intel corei5-6400 2.7 GHz 的 CPU、内存 8 GB 的 Windows 平台下使用 C++ 实现算法。由于本文算法在 CPU 端具有很好的并行性, 在实际编码过程中利用 Intel TBB 技术对算法进行加速, 其在求取分辨率为 640 像素 × 480 像素, 视差等级为 30 的立体视觉对的视差时, 速度可达 30 frame/s。实验选取 Middlebury 测试集中的 Tsukuba、Venus、Teddy、Cones 作为算法测试集, 将误匹配率作为测试指标。同时, 将本文算法的实验结果与 ST 算法<sup>[12]</sup>、MST 算法<sup>[11]</sup>、ARW 算法<sup>[15]</sup>进行比较。为验证本文算法在实际场景中的处理效果及其时间优越性, 还对自然场景下拍摄的分辨率为 640 像素 × 480 像素、最大视差为 30 的立体视觉对进行测试。

在利用 Middlebury 测试集进行评测时, 选取误差阈值为 1 个像素的非遮挡区域 nonocc 进行比较, 结果如表 1 所示。由表 1 可以看出, 在立体匹配时, 本文算法的误匹配率稍差于 MST 算法和 ST 算法。这是由于本文算法的代价聚集基于超像素进行, 而 MST 算法和 ST 算法则以像素为基础进行代价聚集。与 ARW 算法相比, 本文算法的误匹配率较优。这是因为本文算法利用树形滤波器对超像素进行代价聚集, 而 ARW 算法以随机游走方式进行代价聚集。同时, 本文算法包含良好的分割信息, 相较于 ARW 算法, 其保边性能更好, 因此评价系数的提升较为显著。

表 1 4 种算法误匹配率对比 %

测试集	ST 算法	MST 算法	ARW 算法	本文算法
Tsukuba	1.25	1.47	4.10	1.52
Venus	0.20	0.25	2.28	0.36
Teddy	6.00	6.01	7.97	6.22
Cones	2.77	2.87	6.79	2.94

本文算法对于 Tsukuba、Venus、Teddy、Cones 测试集生成的视差伪彩图如图 5 所示。



图 5 Middlebury 测试集上的立体匹配结果

与 Middlebury 测试集的立体视觉对相比, 自然场景下拍摄的立体视觉对的处理难度更大。本文算法对于自然场景下拍摄的图片也表现出较好的立体匹配效果, 如图 6 所示。



图 6 自然场景下的立体匹配结果

基于本文算法在自然场景下的优越性能,结合合理的相机选型,可实现精确物体测距、背景虚化等应用。图 7 展示了本文算法用于物体测距和测尺寸的效果。得到的测量结果为:高  $H = 1\,747.66\text{ mm}$ ,与相机的距离  $D = 1\,503.77\text{ mm}$ ,与实际值( $H = 1\,745.00\text{ mm}$ ,  $D = 1\,548.00\text{ mm}$ )较为接近。



图 7 双目测距示例

本文算法在保证处理效果的前提下,使用 TBB 技术对其进行加速。图 8 给出 5 种算法处理立体视觉对的耗时情况,其中,FA 是未经过加速的本文算法,FA-TBB 是采用 TBB 技术加速后的本文算法。

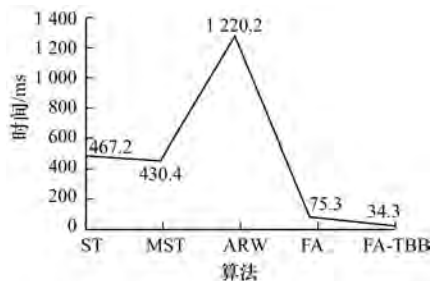


图 8 5 种算法耗时情况对比

由图 8 可以看出,ARW 算法由于使用随机游走进行迭代求解,其速度最慢。ST 算法、MST 算法的

速度可达  $2\text{ frame/s}$ 。而本文算法在未经优化的情况下处理速度为  $13\text{ frame/s}$ ,经过 TBB 加速后可达  $30\text{ frame/s}$ ,其能在保证处理效果的前提下,提高速度。

### 3 结束语

本文提出一种改进的代价聚集快速立体匹配算法。通过 AD-Census 进行初始像素代价计算,并利用 SLIC 进行超像素分割,构建超像素初始代价。以超像素之间的亮度差异为权重生成超像素最小生成树,通过树形滤波器进行超像素代价聚集,再以一定的权重将其与初始像素代价融合。采用 winner take all 进行视差后处理,生成视差图。实验结果表明,该算法能够取得较好的立体匹配效果,大幅减小时间代价。下一步将考虑用递归非全局保持滤波器改进超像素代价聚集过程,加快算法速度。

### 参考文献

- [1] SUN Jian, SHUM Heung Yeung, ZHENG Nanning. Stereo matching using belief propagation [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(7): 787-800.
- [2] HIRSCHMULLER H. Accurate and efficient stereo processing by semi-global matching and mutual information [C]//Proceedings of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Computer Society, 2005: 807-814.
- [3] 王新艳, 潘巍, 王月莲, 等. 基于全局差错能量函数的立体匹配算法 [J]. 计算机工程, 2017, 43(7): 244-249.
- [4] 王年, 范益政, 鲍文霞, 等. 基于图割的图像匹配算法 [J]. 电子学报, 2006, 34(2): 232-236.
- [5] 王召月, 陈丽芳. 基于 mean-shift 全局立体匹配方法 [J]. 计算机工程与科学, 2017, 39(7): 1333-1337.
- [6] HONG Li, CHEN George. Segment-based stereo matching using graph cuts [C]//Proceedings of 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Computer Society, 2004: 74-81.
- [7] 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.
- [8] PORIKLI F. Constant time  $O(1)$  bilateral filtering [C]//Proceedings of International Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2008: 1-8.
- [9] HE Kaiming, SUN Jian, TANG Xiao'ou. Guided image filtering [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(6): 1397-1409.

(下转第 281 页)

(上接第 272 页)

- [10] LU Jiangbo, SHI Keyang, MIN Dongbo, et al. Cross-based local multipoint filtering [C]//Proceedings of International Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2012:430-437.
- [11] YANG Qingxiong. A non-local cost aggregation method for stereo matching [C]//Proceedings of International Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2012:1402-1409.
- [12] MEI Xing, SUN Xun, DONG Weiming, et al. Segment-tree based cost aggregation for stereo matching [C]//Proceedings of International Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2013:313-320.
- [13] ZBONTAR 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.
- [14] LI Lincheng, YU Xin, ZHANG Shunli, et al. 3D cost aggregation with multiple minimum spanning trees for stereo matching[J]. Applied Optics, 2017, 56(12):3411-3420.
- [15] LEE S, LEE J H, LIM J, et al. Robust stereo matching using adaptive random walk with restart algorithm[J]. Image and Vision Computing, 2015, 37:1-11.
- [16] ACHANTA R, SHAJI A, SMITH K, Et al. SLIC superpixels compared to state-of-the-art superpixel methods[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(11):2274-2282.
- [17] MEI Xing, SUN Xun, ZHOU Mingcai, et al. On building an accurate stereo matching system on graphics hardware[C]//Proceedings of International Conference on Computer Vision Workshops. Washington D. C., USA: IEEE Press, 2011:467-474.

编辑 樊丽娜