

doi: 10.3969/j.issn.0490-6756.2018.05.009

基于图像片段匹配的立体匹配算法

李迎江^{1,2}, 张建伟¹, 钟羽中¹, 伍少梅¹

(1. 四川大学计算机学院, 成都 610065; 2. 红河学院, 蒙自 661100)

摘要: 提出一种以图像片段为单位进行立体匹配的方法, 与传统的以像素为单位的立体方法不同的是, 本文使用图像的边缘信息将参考图像分成小的片段, 同时在水平和垂直方向上以小的片段为单位进行立体匹配, 然后进行视差图像的融合, 提高了匹配的准确性. 在片段内, 本文将匹配代价分为高代价与低代价, 以不同的计算方式统计两种代价值对整个片段的代价贡献, 这种计算方式可以准确地匹配物体被部分遮挡时的情况; 接下来对原参考图像进行低位数放大, 对片段内的视差值根据低位数放大后的参考图像再次查找精确视差; 最后, 依据原始的参考图像, 使用同色域均值滤波对获得的视差图进行填补与平滑操作. 经实验证明, 本方法具有算法运算复杂度低, 速度快, 匹配准确和物体的边缘视差清晰的特点, 在 Middlebury 图像库的测试中获得了较好的实验结果.

关键词: 立体匹配; 片段匹配; 图像边缘; 同色域均值滤波

中图分类号: TP751 **文献标识码:** A **文章编号:** 0490-6756(2018)05-0955-10

A stereo matching method based on fragment matching of image

LI Ying-Jiang^{1,2}, ZHANG Jian-Wei¹, ZHONG Yu-Zhong¹, WU Shao-Mei¹

(1. College of Computer Science, Sichuan University, Chengdu 610065, China; 2. Honghe University, Mengzi 661100, China)

Abstract: A method of stereo matching using each image fragments as a unit is proposed. Different from the conventional pixel-by-pixel stereo method, the authors use the edge information of the image to divide the reference image into small segments and perform stereo matching in units of small segments in the horizontal and vertical directions, and then use the disparity images to improve the accuracy of the matching. In each segment, the authors divide the matching cost into high cost and low cost and calculate the cost contribution of the two cost values to the entire segment in different calculation modes. This calculation method can accurately match the situation when the object is partially occluded. Next, the authors enlarge the low-order digit of the intensity for the two reference images and search for more accurate disparity values in all the fragments according to the enlarged reference images. Finally, the disparity map is filled and smoothed using the same color region median filter according to the original reference image. The results of the experiments show that this method has the characteristics of low computational complexity, fast speed, accurate matching, and clear edge disparity of the object. It achieved good performance with the Middlebury stereo benchmark.

Keywords: Stereo matching; Fragment matching; Image edge; Same color region median filter

收稿日期: 2017-11-14

基金项目: 四川省科技厅基金项目(2018RZ0080); 四川大学研究生科研创新基金(2018YJSY073)

作者简介: 李迎江(1980-), 男, 云南蒙自人, 博士生, 研究方向为图像处理与机器视觉. E-mail: 45487743@qq.com

通讯作者: 伍少梅. E-mail: Wu_scdx@126.com

1 引言

在计算机视觉中根据两幅图像生成稠密的对应是非常重要的一个问题^[1]. 特别是在双目立体视觉中, 这个问题就是要据左右相机的两帧图像建立立体匹配^[2]. 这是近年来被广泛研究的一个主题, 由于用来匹配的两幅图像来自不同的视角, 由此产生的遮挡, 光照不同, 还有图像本身的低纹理区域和重复纹理区域, 都给准确的立体匹配带来困难. 为了克服以上困难, 以便生成更准确的立体匹配视差图, 许多研究者做了大量的研究工作.

通常立体匹配算法包含四个步骤: 匹配代价计算、代价聚合、视差计算和视差细化^[2]. 前两步是计算匹配代价, 后两步是计算立体视差^[3], 如何准确高效地计算匹配代价是立体匹配算法的关键问题. 传统方法中, 大多数代价聚合方法可以看作是在代价集上的联合滤波^[4], 如果用简单的线性滤波器, 如盒或高斯滤波器进行代价聚合, 会导致深度边界的模糊^[2], 因此, 采用边缘保护滤波器进行代价聚合成为必然, 如采用双边滤波器^[5-7]和引导滤波器^[8,9]都取得了较好的效果. Yang^[10]提出一种非局部的代价聚合方法, 通过计算图表(Image Graph)的最小生成树(Minimum Spanning Tree, MST)来快速执行非局部代价聚合, 巧妙的使用MST树来计算所有点的代价聚合, 提高匹配精度的同时降低运算复杂度. Mei等人^[11]通过在分段表来生成MST, 可以快速地获得更好的视差图.

最近, 一些研究者开始使用卷积神经网络(Convolutional Neural Network, CNN)来获得更准确的视差图, 其方法是通过真实视差已知的小图像块来训练CNN, 用网络的输出来获取初始的匹配代价^[12-14]. Christos等人^[15]利用硬件实现了基于引导滤波的立体匹配系统, 使立体匹配通过硬件加速在移动设备上实现了实时运行并且有较高的匹配精度. Zhang等人^[9]提出新的视图内插的全局立体模型来计算视差图, 表现出了优秀的性能. Zhan等人^[16]使用经过平滑滤波后的图像作为引导图像的方法来改进代价计算方法, 提高了算法的匹配精度. Li等人^[12]提出了一种称为基于Patch-Match的超像素剪切(PatchMatch Superpixel Cut, PMSC)的新算法, 用来估计像素3D标签的方法实现了更准确的立体匹配, 目前在Middlebury 3.0基准^[17]评价中排名很高.

2 片段匹配方法

本文提出一种新的立体匹配算法, 与传统的立体匹配方法不同, 与基于块分割^[12]和基于特征点的匹配方法也不同. 本文的方法是利用图像边缘信息将图像分成长短不一的图像片段, 并以这些图像片段为匹配单位而不是单个像素来进行立体匹配, 即左参考图中的一个图像片段, 在右参考图的可能视差范围内寻找匹配代价最小的(最相似的)图像片段, 整个匹配代价计算过程不需要代价聚合.

为了提高匹配的准确性, 本文同时在水平方向和垂直方向上对图像的片段进行匹配. 这样做是因为有的图像在水平的方向上的纹理较丰富而垂直方向的纹理较少, 而有的图片在垂直方向纹理丰富而水平方向的纹理较少, 使用水平加垂直方向的片段匹配可以形成互补.

另外, 为了提高算法对于被遮挡物体的准确匹配, 在算法中对片段中的像素的匹配代价进行分类: 低匹配代价和高匹配代价. 匹配时以低匹配代价像素为主, 将高匹配代价的像素点看作非匹配点或者遮挡点, 并赋值为一个合适的影响参数, 最后对所有的匹配代价求和, 实现了对被遮挡物体较精确的匹配.

得到以片段为单位的立体匹配视差图后, 还要对视差进一步求精, 本文对图像的强度值进行尾数放大, 这样可以增强低纹理区域的强度差异. 然后对片段内的每个像素在片段视差值的基础上进行小范围求精. 最后对视差图进行漏洞填补与平滑处理得到最终的视差结果.

综上所述, 本文提出了一种新的立体匹配方法: 基于图像片段匹配的立体匹配方法, 先以水平和垂直方向的图像片段为单位进行初步立体匹配, 然后再对片段内的像素点进行精确匹配. 与传统的匹配方法相比, 匹配速度快, 低纹理区域的匹配更准确, 且物体边缘的视差清晰.

在本文算法中, 将匹配代价进行分类并使用不同的汇总方式, 使算法对遮挡区域的视差估计更加准确; 对视差图的完善过程中, 使用同色域均值滤波对视差图进行填补和去噪, 填补的视差准确且不影响图像边缘.

3 提出的立体匹配方法

本文所提出的基于片段立体匹配方法(Fragment-Based Stereo Matching, FBSM)是以图像的

片段为单位进行立体匹配的,因此,先要对图像进行边缘查找,得到图像的边缘信息.由于一些图像的局部区域纹理过于复杂,左右参考图因光照而产生强度不同,在查找边缘时应先对图像进行预处理,去掉一些弱纹理和图像强度值的干扰.接下来,参照边缘信息,在原始图像上截取图像片段进行立体匹配,立体匹配分为水平立体匹配和垂直立体匹配,最后,进行匹配融合和精炼操作,总体的流程图如图1所示.

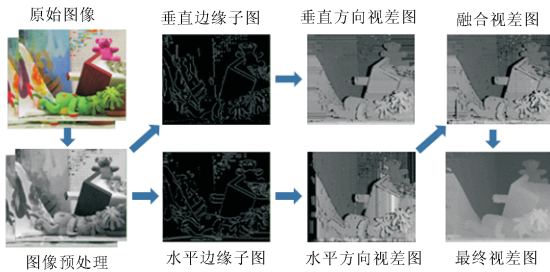


图1 本文算法的流程图

Fig. 1 Flow chart of this algorithm

3.1 参考图像的预处理

本文在图像的预处理阶段主要解决两个问题,(1)是左右参考图像对因光线不同而造成强度差异,这会对后面的立体匹配产生影响,本文对图像对的强度值进行调整,降低光线不同产生的差异;(2)对于一些图像中存在纹理非常复杂的区域,要进行模糊处理,去掉一些弱纹理和复杂区域的纹理,降低这些纹理对边缘查找的影响.

3.1.1 强度调整 首先,对左右两幅参考图像进行强度均衡化操作,减少因光照不同而产生的局部过亮或者局部过暗,然后计算整幅图像的平均强度值,调整时,以强度高的图像为基准,将强度低的图像调高.考虑到相机摆放位置不同而造成的图像内容的差异,在比较两幅图像的平均强度时,本文以两幅图像的共有图像区域为调整参考,去掉图像左右两边的图像带,图像带的大小设置为图像对的最大视差值 d_{\max} ,调整的方法如式(1)所示.

$$\begin{cases} I_L(i, j) = I_L(i, j) \times \frac{ave_{r_R}}{ave_{r_L}}, & \text{if } ave_{r_R} > ave_{r_L} \\ I_R(i, j) = I_R(i, j) \times \frac{ave_{r_L}}{ave_{r_R}}, & \text{if } ave_{r_R} < ave_{r_L} \end{cases} \quad (1)$$

其中, ave_{r_L} 和 ave_{r_R} 表示去掉图像带后的左右参考图的平均强度.

3.1.2 图像模糊处理 对图像进行模糊处理的方法有很多种,这里采用引导滤波^[18]的方法.首先,

本文将灰度参考图进行引导滤波,去掉图像中的弱纹理.但是图像中依然有纹理较强的复杂纹理区域没有被模糊化,因此,我们需要查找这些复杂的强纹理区域,针对这些区域进行模糊化处理.查找方法是:在小窗口中计算所有像素的梯度值,如果梯度值大于某一阈值的像素个数超过窗口总像素个数的一半以上,则对此窗口进行模糊处理.

$$\text{count}(\omega_{i,j} | G_{i,j} > \sigma_g) > n_w / 2 \quad (2)$$

其中, $\text{count}(\cdot)$ 表示对窗口内满足条件的像素统计个数; $\omega_{i,j}$ 为窗口内坐标为 (i, j) 的像素; $G_{i,j}$ 为对应像素的梯度值; σ_g 为梯度阈值.这里的梯度包括水平和垂直两个方向的梯度,式中仅对梯度大于 σ_g 的像素统计个数; n_w 为窗口中的像素个数之和.考虑到算法的容错性和运算效率,本文将窗口尺寸设为 5×5 .以 Middlebury 图像库中的 Teddy 为例,进行模糊处理的结果如图2所示,因为这里的模糊化不需要边缘保护,所以采用高斯滤波进行模糊处理(高斯核大小为 7×7 , 标准差为 0).

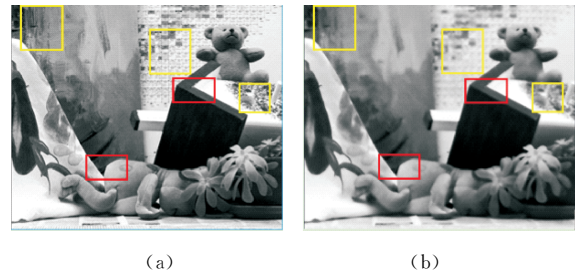


图2 使用本文方法对 Teddy 图像进行模糊处理 (a) Teddy 灰度图; (b) 进行引导滤波和局部高斯滤波后的图像; 黄色方框标示出纹理区域得到模糊处理, 红色方框标示的重要边缘区域保持良好.

Fig. 2 The method in this paper is used to blur teddy images

(a) The teddy gray scale image; (b) The image after guided filtering and local Gaussian filtering; The yellow box indicates that the texture area is blurred, and the important edge area indicated by the red box remains good.

3.2 提取边缘

边缘检测目前常用的方法都是基于图像的梯度信息,如 Sobel, Prewitt, Roberts, LoG^[19] 和 Canny^[20] 等检测算子.这里本文选择使用 Canny 算子进行边缘检测,这是因为 Canny 边缘检测抗噪声能力强,还可以检测到弱边缘.

由于我们要对参考图像在水平和垂直方向分别进行立体匹配,所以,在获取到图像的边缘图像后,要将边缘图像分成水平和垂直方向两个子边缘图像,水平方向子边缘图只保留垂直方向的边缘,而垂直方向子边缘图只保留水平方向边缘.这里本文将总的边缘图像定义为 $Edge_T$, 水平方向子边缘

图像定义为 $Edge_H$, 垂直方向子边缘图像定义为 $Edge_v$.

对在步骤 A 中处理过的 Teddy, 使用 Canny 算法进行边缘提取和分解为水平、垂直方向子图后的结果如图 3 所示.

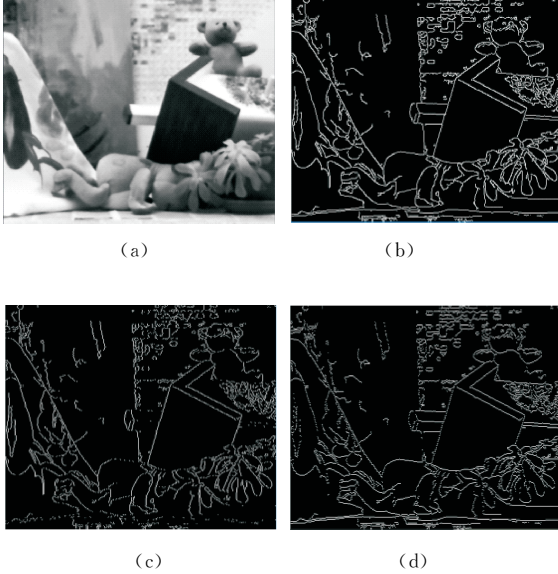


图 3 对 Teddy 图像获取边缘

(a) 步骤 3.1 中处理后的 Teddy 图像; (b) 获取的边缘图; (c) 水平方向边缘子图; (d) 垂直方向边缘子图

Fig. 3 Acquire edges for teddy images

(a) The teddy image processed in step 3.1; (b) The acquired edge image; (c) The horizontal edge sub-image and (d) The vertical edge sub-image

3.3 基于片段的立体匹配

在得到左参考图像的水平 and 垂直方向边缘图后, 就可以在水平和垂直方向进行立体匹配了. 与超像素分割^[21]的立体匹配方法不同, 对于图像片段可以直接进行匹配运算, 而不用对图像块进行学习、训练和识别的过程. 本文的图像片段是这样获取的: 在提取的边缘图像中, 以边缘坐标为起始点, 从水平(垂直)方向上逐行(逐列)进行扫描, 将两个相邻的边缘点 $(x, y_1), (x, y_2)$ 之间的内容看作一个片段 S_{y_1, y_2} , 在原图像中截取这个片段用于本文算法的片段匹配.

3.3.1 水平方向立体匹配 片段 L 的匹配代价是指片段中所有像素的匹配代价之和, 每一个像素的匹配代价这里本文使用灰度加梯度. 匹配代价的基本公式可以被定义为如下形式:

$$E_{cost} = \alpha \cdot cost_v + \beta \cdot cost_g \tag{3}$$

其中, $cost_v = \sum |v_i - v'_i|, i \in L$, 表示整个片段的强度值匹配代价; v_i 为匹配片段 L 上的像素点的灰度值; v'_i 为右参考图上被检测片段上的对应像

素的灰度值; $cost_g = \sum |g_i - g'_i|, i \in L$, 表示整个片段的梯度值匹配代价, g_i 与 g'_i 分别是匹配片段与被检测片段上对应像素点的梯度值; L 为当前片段, α 和 β 分别为灰度匹配代价与梯度匹配代价的权重系数.

当一个物体在左右两幅参考图像对上没有被遮挡时, 基于片段的匹配是简单的, 使用式(3)就可以正确的匹配, 但是, 如果物体的一部分被遮挡, 使用式(3)计算匹配代价就会使匹配结果出现偏移甚至是完全错误的匹配. 因此, 在计算总匹配代价时, 本文将一个片段中所有像素的匹配代价分成两部分, 即低匹配代价和高匹配代价. 其中低匹配代价部分代表与候选匹配片段非常相似的部分, 而高匹配代价我们则认为与候选匹配片段不相似的部分或者片段被遮挡的部分, 对高匹配代价部分不能简单的进行统计, 否则也会出现误匹配. 如图 4 所示, 物体被遮挡时可能出现的误匹配.

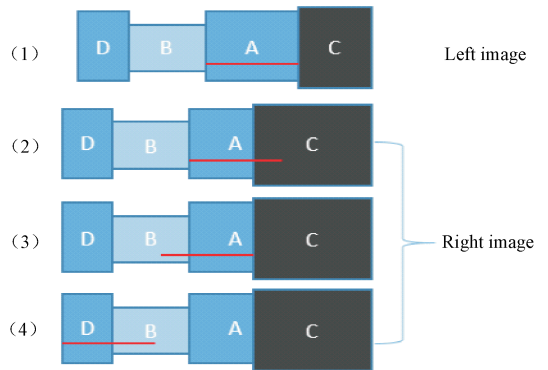


图 4 物体被遮挡的一种可能的情况

(1) 为左参考图; (2)~(4) 为右参考图, 其中(2)为正确的片段匹配, (3)和(4)为可能的错误匹配

Fig. 4 A possible situation in which an object is occluded

(1) The left reference image; (2)~(4) The right reference image, wherein (2) is the correct segment matching, and (3) and (4) are possible error matching

因此, 为了提高对对遮挡物体的准确匹配, 本文需要对式(3)进行改进, 按片段中每个像素的匹配代价与像素本身灰度值与梯度值的比值, 将匹配代价分为低匹配代价点和高匹配代价点. 其中低匹配代价点认为是片段中的正确匹配点, 高匹配代价点认为是遮挡点或错误匹配点. 计算一个片段总的匹配代价 E_{cost} 是以低匹配代价作为片段匹配的主要依据, 而高匹配代价点都用低匹配代价点的平均值乘以一个系数来代替. 将总匹配代价 E_{cost} 最低的片段作为正确的匹配片段, 改进后的片段匹配代价

计算公式为

$$E_{cost} = \alpha \cdot (cost_{vlow} + cost_{vhigh}) + \beta \cdot (cost_{glow} + cost_{ghigh}) \quad (4)$$

其中, $cost_{vlow} = \sum |v_i - v'_i|$, if $(v_i - v'_i) < v_i \times \theta$, $i \in L$, θ 为高、低匹配代价分类的参数; $cost_{vhigh} = count(abs(v_i - v'_i)) \times \overline{cost}_{vlow} \times \delta$, if $(v_i - v'_i) \geq v_i \times \theta$, $i \in L$; \overline{cost}_{vlow} 为低匹配代价的平均值; δ 为相乘系数; $count(\cdot)$ 表示求个数. 这样计算, 可以让高匹配代价的值随着低匹配代价的均值的变化而变化. $cost_{glow}$ 与 $cost_{ghigh}$ 分别是低、高匹配代价的梯度和, 具体计算方法与 $cost_g$ 的计算相似, 这里不再详细描述.

视差值的确定 找到正确的匹配片段后, 就可以把匹配代价保存到视差矩阵中了. 整个视差图矩阵是由所有的片段的视差值构成的. 另外, 在计算视差的同时建立了一个视差置信矩阵, 用来保存每个像素点在匹配时的匹配代价, 为后期的视差融合提供支撑信息. 视差置信矩阵定义如式(6)所示.

$$dis_H(i, j) = \begin{cases} L_{left} - L'_{left}, & \text{if } (I_L(i, j) - I_R(i, j) < I_L(i, j) \times \Delta) \\ 0, & \text{if } (I_L(i, j) - I_R(i, j) \geq I_L(i, j) \times \Delta) \end{cases} \quad (5)$$

$$conf_H(i, j) = \begin{cases} 1/|I_L(i, j) - I_R(i, j)|, & \text{if } (dis_H(i, j) \neq 0) \\ 0, & \text{if } (dis_H(i, j) = 0) \end{cases} \quad (6)$$

其中, $dis_H(i, j)$ 表示水平方向的视差图在坐标点 (i, j) 的视差值; L_{left} 和 L'_{left} 分别为对应当前片段与匹配片段左端点的列坐标; $I_L(i, j)$ 和 $I_R(i, j)$ 分别为左右参考图像在坐标点 (i, j) 的灰度值; Δ 作为一个参数用来判断对应像素点是否真正匹配; $conf_H(i, j)$ 表示水平方向视差图在坐标点 (i, j) 的置信度, 用于后面与垂直方向视差图作交叉验证.

3.3.2 垂直方向立体匹配 由于有些图像的某些区域在垂直方向难以找到边缘, 而在水平方向却容易找到, 这时就可以用垂直方向的片段进行立体匹配, 这可以很好地补充水平方向立体匹配的不足, 也可以对水平方向立体匹配进行交叉验证. 垂直方向的立体匹配方法与水平匹配的方法大致相同, 根据垂直边缘子图找到垂直方向的片段, 然后与右参考图进行立体匹配. 片段的匹配代价计算与式(4)相同, 垂直方向立体匹配获得的视差图和置信度矩

阵表示如下.

$$dis_V(i, j) = \begin{cases} L_{up} - L'_{up}, & \text{if } (I_L(i, j) - I_R(i, j) < I_L(i, j) \times \Delta) \\ 0, & \text{if } (I_L(i, j) - I_R(i, j) \geq I_L(i, j) \times \Delta) \end{cases} \quad (7)$$

$$conf_V(i, j) = \begin{cases} 1/|I_L(i, j) - I_R(i, j)|, & \text{如果 } (dis_V(i, j) \neq 0) \\ 0, & \text{如果 } (dis_V(i, j) = 0) \end{cases} \quad (8)$$

其中, $dis_V(i, j)$ 与 $conf_V(i, j)$ 分别是垂直方向的视差图在坐标点 (i, j) 的视差值和置信度; L_{up} 和 L'_{up} 分别为对应匹配片段上端点的列坐标. 其它参数与式(5)和式(6)是一致的. 下面也以 Teddy 图为例来计算水平和垂直方向的视差图, 如图 5 所示.

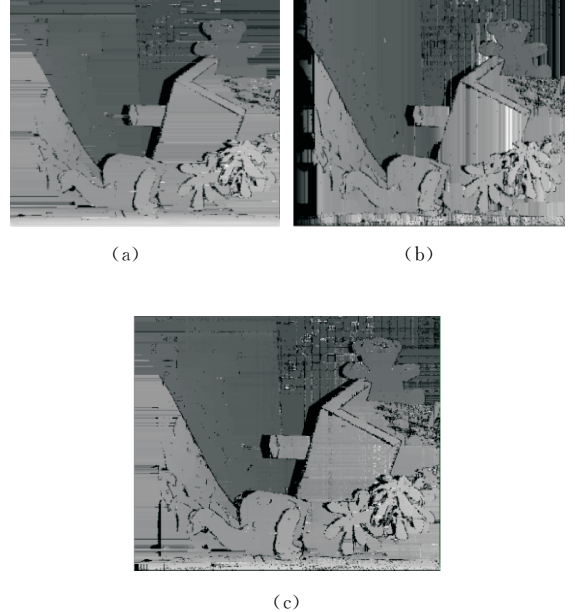


图 5 对 Teddy 图像计算水平和垂直方向的视差图并进行融合

(a) 水平方向计算得到的视差图; (b) 垂直方向计算得到的视差图; (c) 融合后的视差图

Fig. 5 Calculating horizontal and vertical disparity maps of teddy images and fusing them (a) The disparity map in the horizontal direction and (b) The disparity in vertical direction; (c) The disparity map after fusion

3.3.3 视差融合验证 在完成水平与垂直方向的视差计算后, 就可以根据水平和垂直方向获得的视差值矩阵和置信度矩阵进行交叉验证与融合, 以获取更准确的视差结果. 在交叉验证与融合过程中, 本文是以水平方向的视差为主要参考, 以垂直方向获得的视差作为补充验证, 当垂直方向视差置信度大于水平方向的置信度, 并且垂直方向的视差值在

邻域视差窗口内更接近窗口内视差的均值,则用垂直方向视差代替水平方向视差,否则等于水平方向视差.交叉验证的公式如下.

$$\text{disp}_F(i, j) = \begin{cases} \text{dis}_H(i, j), & \text{如果}(\text{conf}_H(i, j) > \text{conf}_V(i, j) \\ \text{and } \text{dis}_H(i, j) \approx \text{dis}_{WH}) \\ \text{dis}_V(i, j), & \text{否则} \end{cases} \quad (9)$$

这里, $\text{disp}_F(\cdot)$ 就是得到的最终视差图矩阵; $\text{conf}_H(\cdot)$ 与 $\text{conf}_V(\cdot)$ 分别为水平和垂直方向的置信度矩阵,交叉验证融合后的视差图如图 5(c) 所示.

3.3.4 片段内精确匹配 以片段为单位进行立体匹配后,本文需要计算片段内每个像素的具体视差值了.首先,利用同一片段内的视差信息一致性的属性对融合后的视差进行去除离群值操作,去掉错误的视差值.然后将左右参考图像的灰度图像进行强度值尾数放大,目的是增大相似像素间的强度差异.放大强度值有些类似于锐化操作,我们将一个非边缘区域的像素点的强度值进行尾数放大,放大方法是个位数放大 5 倍,十位数放大 2 倍,而百位数去掉,如式(10).Mod 表示整除取余数,公式中的除法也是整除, $D_{L,R}(x, y)$ 为尾数放大后的左右参考图.接下来根据以片段获取的视差图 disp_F ,在放大后的左右灰度图中寻找最匹配的点,寻找方法如式(11)所示.

$$D_{L,R}(x, y) = \frac{I_{L,R}(x, y) \text{Mod } 100}{10} * 2 + I_{L,R}(x, y) \text{Mod } 10 * 5 \quad (10)$$

$$\text{disp}(x, y) = t, \text{ if } |D_L(x, y) - D_R(x, y - \text{disp}(x, y))| > |D_L(x, y) - D_R(x, y - t)| \quad (11)$$

其中, $\text{disp}(x, y)$ 为视差矩阵 disp_F 中的一点, $|t - \text{disp}(x, y)| < \omega$, t 是片段内查找到的精确视差值, ω 为视差的查找范围.这里本文仅对片段长度大于 5 的片段进行段内的精确匹配,强度尾数放大及片段内视差精确查找的实验结果如图 6 所示.

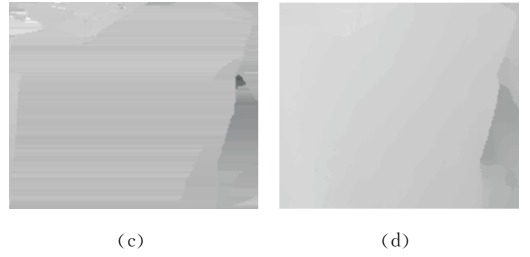


图 6 强度值的尾数放大和片段内的视差精确查找

(a) Adirondack 图片的一部分; (b) 强度值的尾数放大结果; (c) 基于片段求得的视差; (d) 片段内视差精确查找后的结果

Fig. 6 Mantissa amplification of intensity values and accurate searching of disparity within segments

(a) Part of Adirondack image; (b) Amplifying the mantissa of the intensity value; (c) Disparity obtained based on the segment; (d) The result of accurate searching for the disparity in the segment

从图 6 可知,经过片段内视差精确查找后,每个像素点的视差值更加准确了,而且对于整个片段本身匹配不准确的情况,片段内视差精确也能进行校正,校正的范围由参数 ω 决定,当然,设置过大的校正范围会影响程序的运行时间.

4 视差完善

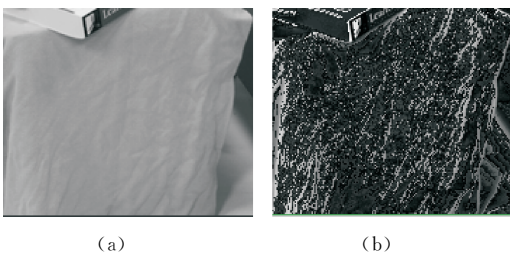
在完成上面的操作之后,已经获得了一个较为准确的视差图了,但是,视差图中因遮挡而造成的空视差区域,以及匹配过程出现小的错误匹配片段,还有因为图像的倾斜表面而产生的非平滑过渡区域,都要做填补和提炼的工作.

①视差值填补:以无视差值的点为中心的一个窗口范围内,根据原参考图像,寻找与无视差点的图像颜色相似的像素在视差图上的视差值,将这些视差值的平均值作为该点的视差值.填补视差值的公式如下:

$$Em(i, j) = \text{ave}(\sum \text{disp}_F(u, v), (u, v) \in W, (\text{disp}_F(u, v) \neq 0, |I_L(u, v) - I_L(i, j)| < \epsilon)) \quad (12)$$

其中, $Em(i, j) \in \text{disp}_F$ 为视差图中值为 0 的点; $\text{ave}(\cdot)$ 表示求平均值; (u, v) 是坐标 (i, j) 滑动窗口 W 内的一个元素; I_L 为原始的左参考图像, ϵ 是用来判断点 (u, v) 与点 (i, j) 在原参考图像上是否相似的参数.

②视差平滑处理:这一步骤主要目的是去噪及平滑视差图.具体的方法与视差填补很相似,对视差图上的每个点,在相邻窗口内找到与窗口中心点属于同一颜色区域(在原参考图 I_L 上进行比对)的



(a)

(b)

视差点, 计算这些视差点的均值, 将此均值作为中心点的值, 这种滤波方法本文称为同色域均值滤波 (SRMF). SRMF 与引导滤波一样, 都需要一幅图像作为参考图, 这里我们是用初始的 I_L 作为相似灰度的参考图, 公式如下

$$\text{dis}p_{\text{SRG}}(x, y) = \text{ave}\{\text{dis}p_{\text{F}}(x-l, y-k), k, l \subset W, \frac{|(I(x, y) - I(x-l, y-k))|}{I(x-l, y-k)} < \sigma\} \quad (14)$$

其中, $\text{dis}p_{\text{SRG}}(x, y)$ 为滤波后的视差图; $\text{dis}p_{\text{F}}(x, y)$ 为滤波前视差图; $I(x, y)$ 为原始视图, 在这里作为判断滑动窗口内的一点与中心点是否为灰度相似点; σ 作为一个参数, 用来判断两个点是否相似. 视差图填补与视差平滑的效果如图 7 所示. 到这里, 立体匹配的操作就全部结束.

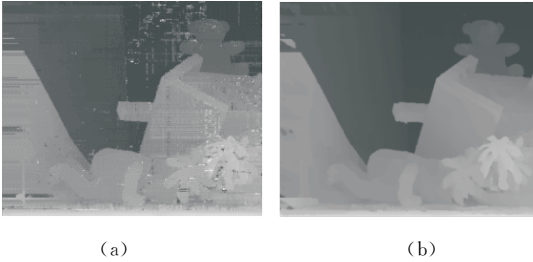


图 7 视差填补与平滑处理

(a) 视差填补后的视差图; (b) 对(a)进行相似灰度中值滤波操作后的视差图

Fig. 7 Disparity filling and smoothing processing
(a) The disparity map after filling; (b) The disparity map after performing similar gray median filtering operation on (a).

5 实验结果

5.1 参数设置

在本文的实验中, 我们应用了 Middlebury 3.0 benchmark^[16], 还有 Middlebury 2003 dataset^[22],

2006 dataset^[23]. 程序是在一台带有 intel i76700 3.4 Hz CPU, 8 G 内存, GTX 960 图像卡的计算机上运行的. 程序中的参数设置上, 灰度匹配代价与梯度匹配代价的权重系数 α 和 β 分别设置为 0.8 和 0.2; 高、低匹配代价分类的参数 θ 设置为 0.2, $\delta = 1.2$, 用来判断对应像素点是否真正匹配的参数 Δ 设置为 0.2; 片段内精确查找时的查找范围参数 ω 设置为 15; 在原参考图像中用来判断领域窗口内图像与中心点图像强度是否相似的参数 $\varepsilon = 0.1$, σ 与 ε 相似, 也设置为 0.1. 上面的参数值在整个实验过程中保持不变.

5.2 Middlebury Benchmark 上的实验结果

本文方法的准确性在 Middlebury 3.0 基准评估. 它有 30 个高分辨率图像对, 为了与其它算法公平比较, 我们使用半分辨率进行基准评估. 这里主要使用基于块匹配算法和传统算法进行比较. 表 1 给出了本文的算法与其它算法在训练集中的平均误差, 其它算法包括: IGSM^[16], SPM-BP^[14], TMAP^[24], CSCA^[25], 实验对比结果如图 8 所示. ‘Bad 0.5~4.0’ 为全分辨率下视差的错误率多于 0.5~4.0 的缩写, ‘Avgerr’ 表示以像素为单位的平均绝对误差, ‘rms’ 是指以像素为单位的均方根误差. ‘A50~A99’ 分别为百分之 50~99 的误差分位数. 本文提出的算法在大部分测试条件下都是排第一的, 具有更好的表现.

从表 1 中可以知, 本文提出的算法在 ‘bad4.0’ 和 ‘avgerr’ 上的表现是最佳的, 这表明本算法在全局的视差上准确率较高, 但是在 ‘bad1.0’ 和 ‘bad2.0’ 上表现不佳, 这说明算法在像素间的视差和细节上仍有不足, 我们可以从图 9 中更明显的看出来.

表 1 本算法与其它 4 种算法在 MiddleburyBenchmark 中测试的错误率. 最好的结果用粗体字表示

Tab.1 The error rate of this algorithm and other four algorithms tested in Middlebury Benchmark. The best results are shown in bold type

Method	bad0.5	bad1.0	bad2.0	bad4.0	avgerr	rms	A50	A90	A95	A99
FBSM	53.9	34.4	18.2	7.71	2.79	5.74	0.75	3.96	14.22	46.3
IGSM	54.5	32.1	18.7	12.6	5.57	20.2	0.71	9.96	23.81	107.0
SPM-BP	61.1	36.9	22.6	12.8	6.25	21.5	1.47	7.74	24.86	118.9
TMAP	58.8	29.2	17.1	10.9	4.75	18.1	0.70	7.54	25.92	90.0
CSCA	82.4	49.6	34.2	15.5	8.87	27.3	2.40	18.72	55.60	172.3

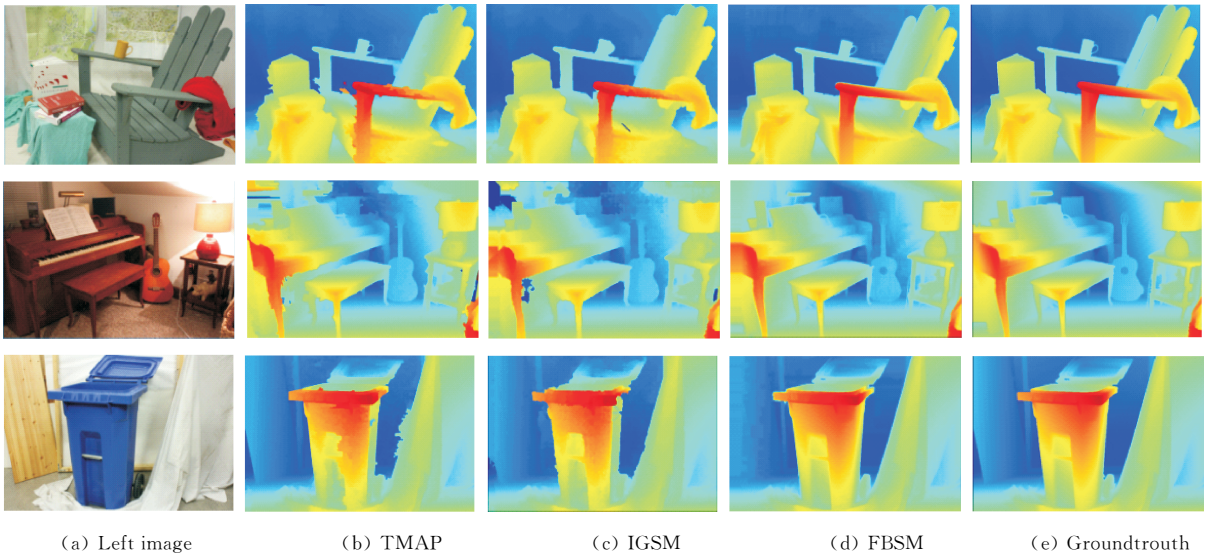


图 8 本文提出的 FBSM 算法与 TMAP 和 IGSM 的视差图与真实视差图的比较
 Fig. 8 Comparison of disparity map and Groundtruth between FBSM algorithm and TMAP and IGSM algorithm

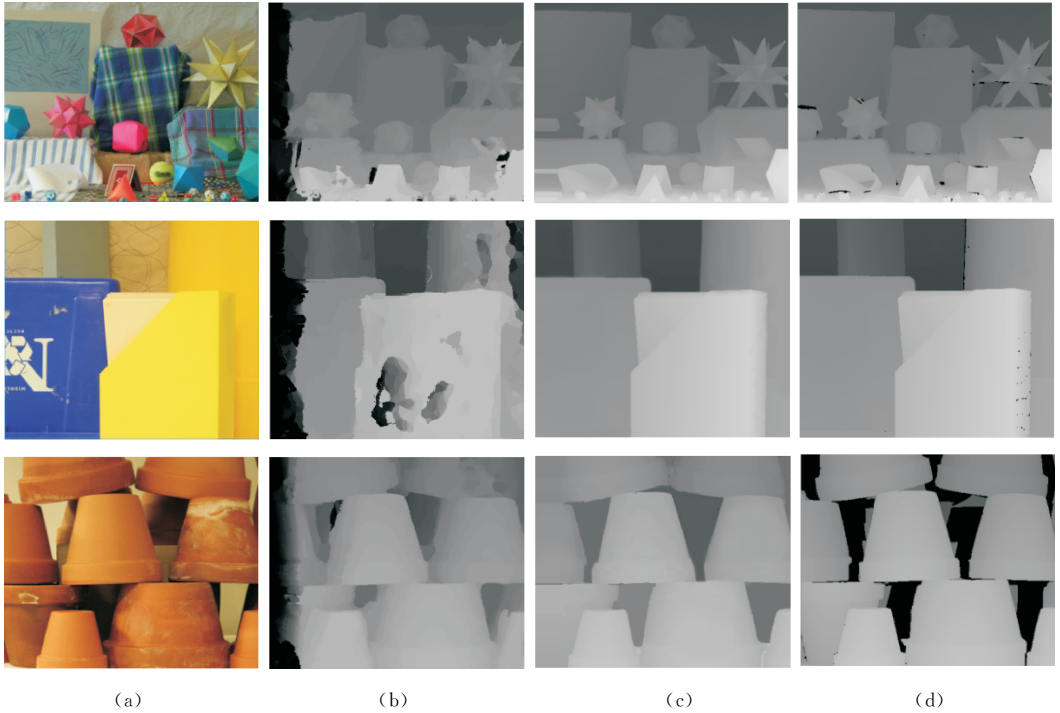


图 9 提出的算法与 CSCA^[25] 算法的视差图比较
 (a)为 Middlebury 数据集中的 Moebius, Plastic, Flowerpots; (b)为 CSCA^[25] 算法获得的视差图; (c)为 FBSM 算法获得的视差图; (d)为 Groundtruth

Fig. 9 Disparity map comparison between the proposed algorithm and CSCA^[25]
 (a) Images of Moebius, Plastic and Flower pots in Middlebury data set; (b) Disparity map obtained by CSCA^[25]; (c) Disparity map obtained by FBSM and (d) Groundtruth.

本文的算法与其它算法结果相比最大的优势在于 rms(即均方根误差)要明显低于其他算法,这表明本文提出的算法在整个图像上的视差偏离是最小的,如果在视差精度要求不是非常严格的情况下,本算法应该是表现最好的。

本算法的另一个优势是对物体的内部低纹理区域的匹配非常准确,算法首先是以片段为单位进

行匹配的,只要片段匹配正确,整个片段内部的偏差就会非常小。如图 9 所示,所提出的算法对于物体内部低纹理或无纹理区域有较高的匹配准确性,特别是对 Plastic 获取的视差图中,本算法与 CSCA^[25] 算法相比表现优异。

5.3 参数和时间分析

这里评估一下本文的算法对于一些关键参数

的敏感性. 参数 θ 用来区分一个片段在匹配时, 其匹配值是高匹配值还是低匹配值, 这个分类决定了一个匹配片段哪一部分是准确匹配, 这是本算法提高对物体被遮挡时的匹配准确率的重要参数. 参数 θ 的取值范围为 $[0, 1]$, θ 越小, 匹配的容错率越低, 算法的准确性会受到极大影响; θ 值越大, 属于低匹配值的越多, 此时算法无法对被遮挡区域进行正常匹配. 如图 10(a) 所示, 我们列出了 Middlebury 标准库中 training 数据集里的 3 组图像对的错误率.

参数 ω 是在片段内精确匹配时, 片段内的每个视差值在多大的范围内进行视差精确匹配, 参数 ω 如果设置较小, 则重新精度匹配的范围就很小, 如果 ω 较大, 则算法将退化为以像素为单位的匹配算法, 而且会使算法运算复杂度上升. 参数的变化与匹配准确性的关系如图 10(b) 所示, 当 $\omega \geq 15$ 时, 错误率变化比较平稳, 这里为了降低算法的时间复杂度, 在不影响算法精度的情况下, 本文选择了 $\omega = 15$.

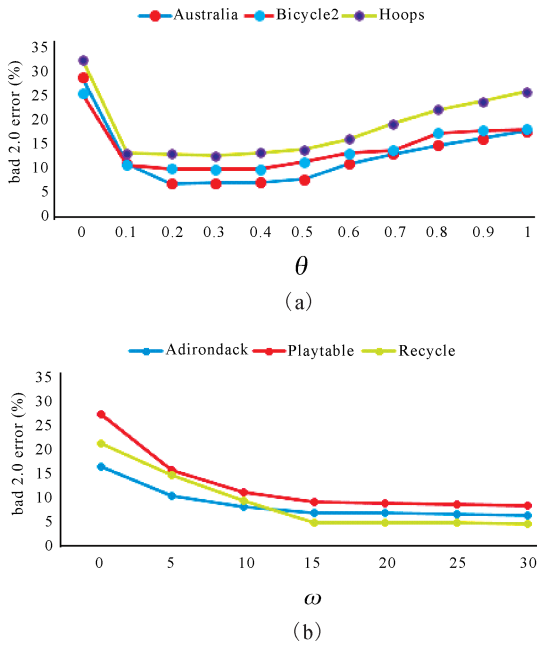


图 10 参数 θ 与 ω 对算法的敏感性分析
 (a) 为参数 θ 对算法错误率的影响; (b) 为参数 ω 对算法错误率的影响
 Fig. 10 Sensitivity analysis of parameters θ and ω to the algorithm
 (a) The influence of parameter θ on the error rate of the algorithm; (b) The influence of parameter ω

这里我们对算法进行一个简单的复杂度分析, 假设原始参考图像对的尺寸为 $W \times H$, 视差范围为 D , 每个步骤的复杂性被表征为: 预处理过程为

$O(nWH)$, 基于片段匹配过程为 $O((2D + 2\omega)WH)$, 视差完善过程为 $O(mWH)$. 其中, n 为对图像进行模糊, 强度调整, 边缘查找和相似区域强度调整所用运算, $n \ll D$, $(2D + 2\omega)$ 为水平和垂直方向的片段和片段内精确查找所用运算; m 为视差值填补与平滑过程所用运算, $m \ll D$, 因此, 整个算法的主要运算集中在匹配过程. 对于传统的基于像素的立体匹配算法, 其运算复杂度在不包括预处理的情况下可表示为: $O(\omega^2(2D)WH)$, 其中, ω 为匹配时使用的窗口尺寸 ($\omega \geq 7$), 这里我们还没有考虑具体的匹配算法每一次匹配的运算量. 因此, 我们可以看出, 本文提出的算法的运算复杂度明显比传统基于像素的立体匹配要低. 在运行时间上, 本文在执行 Middlebury 标准库 training 数据集上的四分之一分辨率的所有图像时用时合计为 455 s, 而 CSCA 算法的执行时间约为 3200 s, IGSM 算法约为 1300 s, 在运行时间上, 本文的算法是运行比较快的.

6 结 论

本文提出了一种基于图像片段的立体匹配算法, 利用图像的边缘信息获得图像的片段, 在水平和垂直方向上分别进行立体匹配, 将两方向获得的视差图进行交叉验证, 并在片段内对获得的视差在一定范围内再次进行精确匹配, 得到更准确的视差图, 最后, 对被遮挡区域的视差进行填补和对非边缘区域的视差值进行平滑操作, 得到最终的视差图像. 实验表明, 本算法运算复杂度低, 获取的视差较准确, 特别是图像的边缘和物体内部匹配准确, 但不足之处在于对倾斜的平面的视差平滑上还需要提高. 总的来说, 本文提出了一种利用图像边缘信息和图像片段匹配的方法进行立体匹配, 并取得了一定的成果, 希望能给其它研究者提供一种思路, 共同推进立体匹配技术的发展.

参考文献:

[1] Liu C, Yuen J, Torralba A. SIFT flow: dense correspondence across scenes and its applications [J]. IEEE Trans Pattern Anal, 2011, 33: 978.
 [2] Scharstein, Daniel, Szeliski, et al. A taxonomy and evaluation of dense two-frame stereo correspondence Algorithms[C]// SMBV. Boston: IEEE Computer Society, 2001.
 [3] Hirschmuller H, Scharstein D. Evaluation of stereo matching costs on images with radiometric differ-

- ences [J]. IEEE Trans Pattern Anal, 2009, 31:1582.
- [4] Rhemann C, Hosni A, Bleyer M, *et al.* Fast cost-volume filtering for visual correspondence and beyond [C]// Proceedings of IEEE Conference on CVPR. Boston: IEEE Computer Society, 2011.
- [5] Tomasi C, Manduchi R. Bilateral filtering for gray and color images[C]// Proceedings of the Sixth International Conference on Computer Vision. Bombay, India: IEEE, 1998.
- [6] Fu L, Peng G, Song W. Histogram-based cost aggregation strategy with joint bilateral filtering for stereo matching [J]. Let Comput Vis, 2016, 10: 173.
- [7] Yang Q. Hardware-efficient bilateral filtering for stereo matching [J]. IEEE Trans Pattern Anal, 2014, 36: 1026.
- [8] Yang Q, Li D, Wang L, *et al.* Full-image guided filtering for fast stereo matching [J]. IEEE Signal Proc Let, 2013, 20: 237.
- [9] Zhang C, Li Z, Cheng Y, *et al.* MeshStereo: A global stereo model with mesh alignment regularization for view interpolation[C]// IEEE ICCV. Boston: IEEE Computer Society, 2015.
- [10] Yang Q. A non-local cost aggregation method for stereo matching [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, [s. l.]: IEEE, 2012.
- [11] Wei X, Sun X, Dong W M, *et al.* Segment-tree based-cost aggregation for stereo matching[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. [s. l.]: IEEE, 2013.
- [12] Li L, Zhang S, Yu X, *et al.* PMSC: Patchmatch-based superpixel cut for accurate stereo matching[J]. IEEE Trans Circ Syst Vid, 2016, 99: 1.
- [13] Bontar J, Lecun Y. Stereo matching by training a convolutional neural network to compare image patches[J]. J Mach Learn Res, 2016, 17: 2287.
- [14] Li Y, Min D, Brown M S, *et al.* SPM-BP: sped-up patchmatchbelief propagation for continuous MRFs [C]// Proceedings of the IEEE International Conference on Computer Vision. [s. l.]: IEEE, 2016.
- [15] Ttofis C, Kyrkou C, Theodoridis T. A low-cost real-time embedded stereo vision system for accurate disparity estimation based on guided image filtering[J]. IEEE Trans Comput, 2016, 65: 2678.
- [16] Zhan Y, Gu Y, Huang K, *et al.* Accurate image-guided stereo matching with efficient matching cost and disparity refinement[J]. IEEE Trans Circ Syst Vid, 2016, 26: 1632.
- [17] Scharstein D, Hirschmüller H, Kitajima Y, *et al.* High-resolution stereo datasets with subpixel-accurate ground truth [M]// Pattern Recognition. Berlin: Springer, 2014.
- [18] He K, Sun J, Tang X. Guided image filtering[M]// Computer Vision-ECCV 2010. Berlin : Springer, 2010.
- [19] Gonzalez R, Woods R. Digital image processing[M]. Beijing: Publishing House of Electronics Industry, 2010.
- [20] Canny J. A computational approach to edge detection [J]. IEEE Trans Pattern Anal, 1986, 8:679.
- [21] Yamaguchi K, Mcallester D, Urtasun R. Efficient joint segmentation, occlusion labeling, stereo and flow estimation[M]// Computer Vision-ECCV 2014. Berlin Heidelberg: Springer International Publishing, 2014.
- [22] Scharstein D, Szeliski R. High-accuracy stereo depth maps using structured light[C]// IEEE Computer Society Conference on CVPR. Boston: IEEE Computer Society, 2003.
- [23] Hirschmuller H, Scharstein D. Evaluation of cost functions for stereo matching[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. [s. l.]: IEEE, 2007.
- [24] Psota E T, Kowalczyk J, Mittek M, *et al.* MAP Disparity estimation using hidden markov trees[C]// Proceedings of IEEE International Conference on CV. [s. l.]: IEEE, 2016.
- [25] Zhang K, Fang Y, Min D, *et al.* Cross-scale cost aggregation for stereo matching [C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. [s. l.]: IEEE, 2014.

引用本文格式:

中文: 李迎江, 张建伟, 钟羽中, 等. 基于图像片段匹配的立体匹配算法[J]. 四川大学学报: 自然科学版, 2018, 55: 955.

英文: Li Y J, Zhang J W, Zhong Y Z, *et al.* A stereo matching method based on fragment matching of image [J]. J Sichuan Univ; Nat Sci Ed, 2018, 55: 955.