

基于贝叶斯推理的多尺度双目匹配方法

曾灿灿, 任明俊, 肖高博, 殷跃红

上海交通大学机械与动力工程学院, 上海 200240

摘要 目前, 虽然有许多双目匹配算法都可以实现较高的匹配精度, 但其中能实现视频级实时计算的却极少。为此, 提出一种基于贝叶斯推理的多尺度优化方法, 在保证算法实时性的同时, 可达到非常高的匹配精度。简单的局部匹配算法在设置不同窗口大小时, 计算出来的视差图会反映不同尺度下场景的结构信息。基于此, 提出一种对多张带有尺度信息和互补性的视差图进行基于贝叶斯推理的联合优化, 从而得到高精度视差图的方法。对 Middlebury 立体视觉数据集的测试结果表明, 本文算法在精度和效率上均优于其他实时算法。

关键词 机器视觉; 立体匹配; 多尺度优化; 实时; 贝叶斯推理

中图分类号 TP391.41 **文献标识码** A

doi: 10.3788/AOS201737.1215004

Multi-Scale Stereo Matching Based on Bayesian Reasoning

Zeng Cancan, Ren Mingjun, Xiao Gaobo, Yin Yuehong

School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Abstract Most of the current stereo matching algorithms have high matching accuracy, but there are very few of them can realize real-time matching with video level frame rate. We present the multi-scale optimization algorithm based on Bayesian reasoning, which can be used to improve the matching accuracy, while maintaining the real-time performance. This algorithm obtains disparity maps with different scales by setting different window sizes. Based on this, the joint optimization based on Bayesian reasoning is proposed to optimize the disparity maps with scale information and complementarity. And then the high precision disparity maps are obtained. Test results of Middlebury stereo vision datasets show that the proposed algorithm has better accuracy and higher efficiency than several real-time algorithms.

Key words machine vision; stereo matching; multi-scale optimization; real-time; Bayesian reasoning

OCIS codes 150.1135; 100.6890; 330.1400

1 引 言

随着移动机器人技术的快速发展和增强现实(AR)、虚拟现实(VR)技术的兴起, 让机器人实时感知三维世界的的需求越来越迫切。实际应用中, 一般要求位置信息的获取速度至少达到 30 frame/s。如何以这样高的速度获取精确的深度图成为前述技术发展中的一项关键技术^[1-2]。目前, 有多种方法可以获取现实世界的三维信息, 如结构光法、飞行时间法、单目运动恢复结构或者双目视觉方法。结构光法可以达到较高的精度, 但标定复杂, 而且使用场合一般限制在室内或一个有限的空间^[3]; 飞行时间法是一种新兴的技术, 利用激光在空中来回的时间差计算距离, 但利用低价格的传感器很难达到较高的精度和分辨率^[4]; 单目相机加上运动相较于双目视觉方法难度更大, 而且只能得到大致的相对距离。因此, 利用双目视觉获取三维信息一直以来都是计算机视觉或机器视觉领域的研究热点^[5-6]。

双目立体视觉通过模拟人的双眼, 利用物体相对左、右相机的视角差异计算出物体和相机的相对位置^[7]。双目立体视觉可以在同一时刻同时获得空间中几十万甚至几百万个点的颜色(灰度)和位置信息, 在

收稿日期: 2017-06-30; **收到修改稿日期:** 2017-08-08

基金项目: 国家自然科学基金(51505404, 51675456)、上海浦江计划(16PJ1404300)

作者简介: 曾灿灿(1993—), 男, 硕士研究生, 主要从事立体视觉方面的研究。E-mail: zcc2015@sjtu.edu.cn

导师简介: 任明俊(1983—), 男, 博士, 讲师, 主要从事机器学习、机器视觉和大规模计算等方面的研究。

E-mail: renmj@sjtu.edu.cn(通信联系人)

一定范围内达到较高的位置精度,但其速度和准确率在很大程度上取决于对左右图像中同一物体进行匹配的算法的速度和精度。因此,双目匹配算法是双目立体视觉的关键点和难点^[5]。双目匹配算法根据模型的复杂度可以分为局部算法、全局算法和半全局算法^[6]。局部算法通过对每一个像素计算其给定不同视差值时的代价,构建整幅图像对应的代价空间,然后直接选取使局部代价最小的视差值作为结果^[8-9]。局部算法目前的主流是在代价聚合阶段利用自适应权值方法对代价空间滤波^[10],在达到稳健性的同时保存图像边缘细节。局部算法非常适合于并行计算,完全可以满足实时性要求,但是其精度较低,受噪声影响严重。全局算法是将逐像素的视差搜索转换成一种全局能量的优化过程,所有点的视差值是同时得到的。通过合理设置匹配代价和不连续代价,全局算法可以达到非常高的精度^[11-13]。当前研究最为广泛的是置信度传播(BP)算法^[14]和图割(GC)算法^[15]。但这类算法的空间复杂度和时间复杂度均非常高,尚难以实际应用。半全局算法是在求解全局算法的全局能量时,利用多方向的动态规划方法求取一个近似全局最优的局部最优解^[16-17]。该算法因为在精度和速度上取得了比较好的平衡,一经提出便受到了广泛的关注,但是这种算法在复杂度上依然高于局部算法,在算法上能并行的程度有限,所以即使在使用图形处理器(GPU)进行加速的情况下也仅能达到近实时的速率^[18]。

目前,实时双目匹配算法研究主要有 2 种思路:1)针对全局和半全局算法;2)针对局部算法。

对全局和半全局算法的改进是通过简化数学模型或者将全局优化拆分成多个局部优化问题,经过迭代逼近全局最优解实现的。Yang 等^[19]提出一种基于快速收敛 BP 的匹配算法。由于标准的 BP 算法速度非常慢,不适合实际应用,为了提高标准 BP 算法的计算效率,该研究采用分层 BP,利用图像金字塔减少最底层(分辨率最高)算法的迭代次数,同时提出了自适应更新每个节点代价的方法以进一步加快收敛。Wang 等^[20]在动态规划(DP)算法的基础上加入自适应代价聚合阶段,并且只在垂直方向上进行代价聚合,从而减少传统动态规划产生的横向条纹问题。该研究利用了 GPU 的并行性和向量处理能力,大大加速了传统的动态规划算法。

对局部算法的改进是在代价聚合阶段利用联合双边滤波或者引导滤波的保边平滑特性对初始代价空间进行滤波,通过聚合的方式提高信噪比从而减少误匹配率,或者是在视差图后处理阶段利用加权中值滤波对由局部算法得到的含有较多噪声的视差图进行保边平滑滤波,从而降低误匹配率。Mattocchia 等^[21]采用双边滤波算法对代价空间进行滤波,之后直接对每个像素求取最小代价对应的视差值。为了提高计算速度,该研究通过对双边滤波窗口分块,降低计算复杂度,又通过 GPU 对算法进行加速,达到了近实时的计算速度。Ma 等^[22]提出了一种加权中值滤波方法来对带有大量噪声的视差图进行后处理。该方法的主要思想是改进中值滤波算法,以在平滑图像的同时保留边缘细节。研究人员先以参考图为指导,使用一个权值核函数来计算中心点周围窗口中各个位置的权值,然后根据这些权值信息获得一个该中心点窗口像素累加得到的加权直方图,再根据加权直方图选择其中的中值作为中心点的最终视差值。该方法的优点在于能较好地保留物体的形状信息,但是计算量非常大。Mattocchia^[23]提出使用局部连续(LC)假设对初始视差图施加约束,利用贝叶斯推理的方式求取使后验概率最大化的视差值。由于该方法的精度依赖于初始视差图的精度,所以不同的初始视差图获取方式会对算法整体的计算速度和精度产生较大的影响。

文献[19-20]对原始的全局和半全局方法做了一定的简化,并利用 GPU 加速得到了实时的计算速度,但其本质上仍然属于全局和半全局方法,因此计算复杂度并没有降低。当图像分辨率提高时,算法的计算耗时也逐渐增加。文献[21-23]均对局部算法做复杂的滤波处理,但算法仍然是局部化的,相对于文献[19-20]中的方法,在计算复杂度上仍然占优。考虑到实际应用中可能会需要处理较高分辨率的图像,本文选择计算复杂度更低的后一类方法。

研究发现,如果在匹配算法中加入多尺度信息会大大提高匹配的准确性。而在现有的双目匹配算法中,运用到多尺度思想的算法大多是利用了图像金字塔来消除高分率辨图像匹配时的歧义性^[19,24-26]。这些算法的思想大多是对图像金字塔的顶层做匹配,将得到的结果代入下一层图像中作为初始值做进一步的匹配。优点是可以在一定程度上消除匹配的歧义性,而且加快全局算法的收敛速度,但弊端是一旦在较低分辨率的匹配中出现误匹配,将很难消除,并且误匹配区域会逐层扩大,最后形成区域性的误匹配,给视差图的后处理增加了难度。其他运用到尺度思想的步骤,如设置窗口大小与图像分辨率成正比,这样的设置在多数情况下

当面对不同分辨率的照片时处理的其实仍是同一个尺度。稀疏图像匹配中利用多尺度特征进行特征点的匹配^[27-28],虽然在一定程度上用到了图像的多尺度信息,但是并没有将不同尺度上的信息进行综合和推理,从而计算出一个最优的结果。

贝叶斯推理是一种基于统计概率的优化方法,稳健性非常好,受噪声等因素的影响较小,并且可以很好地结合图片的内容信息和一些先验知识。基于此,本文利用局部算法的快速性和在设置不同参数时得到结果之间的互补性,借助贝叶斯推理,对多张带有尺度信息的视差图进行联合优化,从而实现实时获取高精度视差图的目的。

2 基本原理

视差空间的多尺度寻优(MSO)指的是利用贝叶斯推理方法对多幅反映不同尺度信息的视差图进行综合、推理,从中找出一组累积似然度最大的视差组合。如图1所示,首先,根据双目图像通过改变匹配算法参数获取一系列视差图,它们反映了场景在不同尺度上的深度信息,部分视差图保留了许多场景的细节,另一部分则反映了场景的整体深度分布情况。这些视差图重叠在一起构成了一个视差空间。之后,以视差图反映的尺度和场景的纹理丰富程度等信息作为条件,对基于局部平滑假设得到的视差图做进一步推理,计算每一点取不同视差值时的累积似然度,累积似然度最大的即是综合了多个尺度信息的最优视差值。

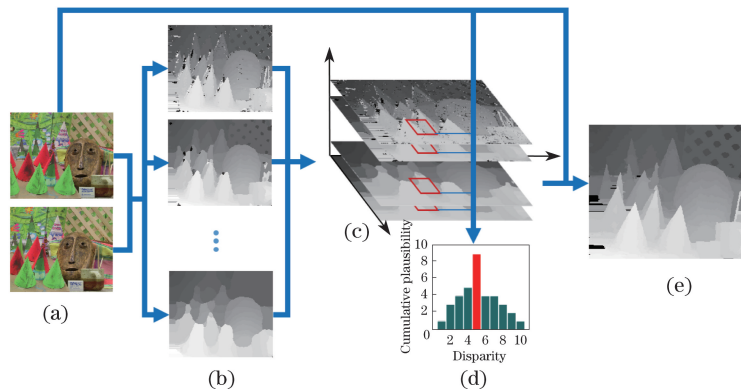


图1 多尺度寻优算法流程图。(a)双目视觉左右图像;(b)带有尺度信息的视差图;(c)视差图构成的视差空间;
(d)某一个像素对应的累积似然度直方图;(e)最终优化结果

Fig. 1 Flow chart of multi-scale optimization algorithm. (a) Left and right images of binocular vision;
(b) disparity maps with scale information; (c) disparity space constructed by disparity maps;
(d) histogram of cumulative plausibility of one pixel; (e) final optimization results

2.1 视差空间获取

本文提出的视差空间优化方法理论上对任何匹配算法的结果都适用,但由于计算效率的原因,建议将这个用于比较简单并且容易并行化实现的局部匹配算法。以最简单的块匹配(BM)为例获取初始的视差图,组成视差空间。BM算法通过比较左右图像中块与块之间的相似度确定图像块的匹配关系,属于局部匹配算法。

根据文献[10]的框架,首先建立一个代价空间,并对代价空间进行滤波,采用胜者为王算法直接选取使聚合代价最小的视差值作为输出结果。初始代价空间中逐像素的匹配代价采用的是像素绝对差值,如果是彩色图像,则是3个通道对应的绝对差值的最大值。对代价空间的滤波采用了可以高度并行且最为简单的盒滤波。图2是对Middlebury立体视觉数据集^[6,29]在代价聚合阶段设置不同窗口大小时得到的结果,其中图2(a)是参考图,从左到右分别是tsukuba、venus、teddy和cones数据集,图2(b)~(d)分别是在不同窗口(窗口直径分别为1/100,1/50和1/30图像宽度)下得到的视差图。

由图2可以看出:当窗口较小时,得到的视差图细节保留更多,但是噪声明显,在部分低纹理区域会得到完全错误的结果;当窗口较大时,视差图噪声明显被抑制,部分低纹理区域也被纠正,却损失了大量的细节,视差不连续的地方错位明显。由此可知,对于简单的局部匹配算法,在代价聚合阶段很难找到一组合适的参

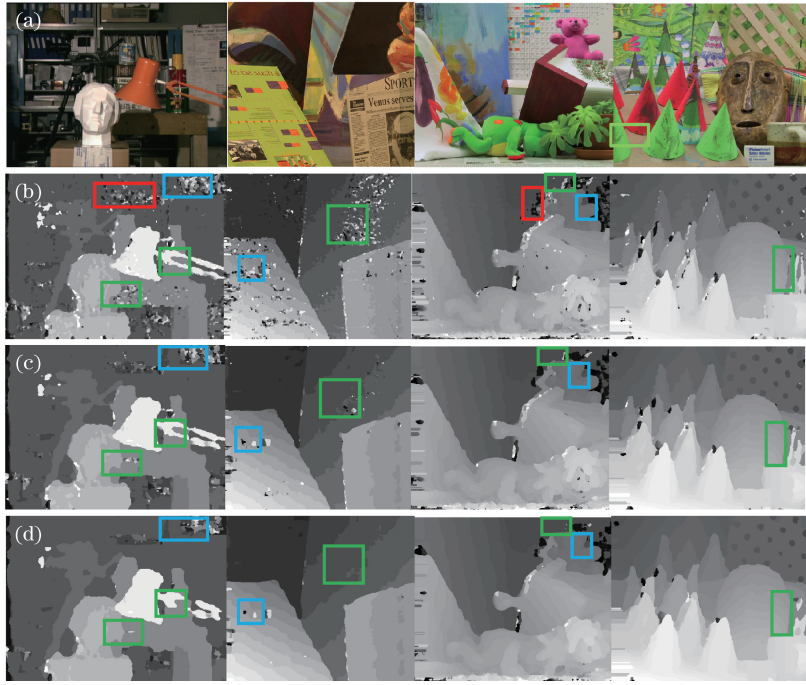


图 2 (a)参考图;不同窗口下得到的视差图:(b)窗口直径为 1/100 图像宽度,(c)窗口直径为 1/50 图像宽度,(d)窗口直径为 1/30 图像宽度

Fig. 2 (a) Reference images; disparity maps with different window sizes: (b) window diameter is 1/100 image width; (c) window diameter is 1/50 image width; (d) window diameter is 1/30 image width

数,使得最终的计算结果可以在获取精细的细节的同时还能正确处理大面积的低纹理区域。

这种现象发生的主要原因是局部匹配算法在进行代价聚合时隐式地应用了一个假设,即在支撑窗口内,所有点的视差值是一致的。这一假设认为被拍摄的场景是局部平滑且前向平行的。这在窗口较小时对大部分的场景都具有较好的适应性,但当窗口变大时,这个假设往往就不成立了,尤其是在具有细小结构的地方。因此,大多数基于支撑窗的局部算法在窗口较小时对于细小结构和物体边缘处有更好的表现,而当窗口较大时,通常在低纹理区域有更好的表现。这是因为低纹理区域往往都是平滑的,符合了代价聚合的局部平滑假设,而且窗口较大时信噪比一般较高,对噪声的影响更加不敏感。

2.2 基于贝叶斯推理的多尺度联合优化方法

先对 2 个最重要的事件 $E_{fg}^{(s)}$ 和 $E_g^{(s)}(d)$ 进行定义^[23]。

$E_{fg}^{(s)}$:点 g 属于 W_f 。其中 f 是参考图中的一个点, g 是 f 的支撑窗中的一个点, W_f 表示以 f 为中心的一个支撑窗, s 代表视差空间中的第 s 层, s 隐含了视差图的尺度信息。这个事件隐含了 g 属于中心在 f 的前向平行平面的概率。假设场景是处处平滑的,那么可以很自然地认为这个事件发生的概率与 g 和 f 的空间相似度是相关的。对于这个事件,假设 f 和 g 的空间距离越小,两者属于同一个前向平行平面的概率越大。

$E_g^{(s)}(d)$:点 g 的视差值为 d 。这个事件隐含的是 g 点处视差为 d 的概率,即视差空间中第 s 层 g 点处视差值为真值的概率。

根据贝叶斯定理可以得到 2 个事件 $E_{fg}^{(s)}$ 和 $E_g^{(s)}(d)$ 的联合后验概率

$$P[E_{fg}^{(s)}, E_g^{(s)}(d) | \Delta_{fg}^{\psi}, \nabla_{R_f}, \nabla_{D_g^{(s)}}] \propto P_P[E_{fg}^{(s)}, E_g^{(s)}(d)] \cdot P_L[\Delta_{fg}^{\psi}, \nabla_{R_f}, \nabla_{D_g^{(s)}} | E_{fg}^{(s)}, E_g^{(s)}(d)], \quad (1)$$

式中 P_P 和 P_L 分别是先验概率和似然函数,函数 Δ_{fg}^{ψ} 用来衡量 f 和 g 的颜色相似度, ∇_{R_f} 表示参考图 R 中点 f 处的平均梯度,它表征了 f 点附近的纹理丰富程度, $\nabla_{D_g^{(s)}}$ 表示视差空间中第 s 层 g 点附近视差平均变化次数。之所以引入这个参数,是因为如果视差图中某些地方视差值变化过于频繁,很可能就是一块误匹配区域,如图 2(b)用红色框框起来的地方。

显然, $E_{fg}^{(s)}$ 和 $E_g^{(s)}(d)$ 事件是相互独立的,为了简化公式,假设 $\Delta_{fg}^{\psi}, \nabla_{R_f}, \nabla_{D_g^{(s)}}$ 3 个事件也是相互独立的,

那么(1)式可以重新写为

$$P[E_{fg}^{(s)}, E_g^{(s)}(d) | \Delta_{fg}^\psi, \nabla_{R_f}, \nabla_{D_g^{(s)}}] \propto P_P[E_{fg}^{(s)}] \cdot P_P[E_g^{(s)}(d)] \cdot P_L[\Delta_{fg}^\psi | E_{fg}^{(s)}, E_g^{(s)}(d)] \cdot P_L[\nabla_{R_f} | E_{fg}^{(s)}, E_g^{(s)}(d)] \cdot P_L[\nabla_{D_g^{(s)}} | E_{fg}^{(s)}, E_g^{(s)}(d)] \propto P_P[E_{fg}^{(s)}] \cdot P_P[E_g^{(s)}(d)] \cdot P_L[\Delta_{fg}^\psi | E_{fg}^{(s)}] \cdot P_L[\nabla_{R_f} | E_g^{(s)}(d)] \cdot P_L[\nabla_{D_g^{(s)}} | E_g^{(s)}(d)]. \quad (2)$$

对于先验概率,假设 $P_P[E_{fg}^{(s)}]$ 与 f, g 之间的欧氏距离 $\Delta_{f,g}$ 的关系为

$$P_P[E_{fg}^{(s)}] = \exp\left(-\frac{\Delta_{f,g}}{\gamma_s}\right), \quad (3)$$

式中参数 γ_s 用来控制空间距离相似度。 $P_P[E_g^{(s)}(d)]$ 由于没有任何先验知识,所以将其设置为一个固定值 1。和许多研究一样^[4,21],将后验概率设置为

$$P_L[\Delta_{fg}^\psi | E_{fg}^{(s)}] = \exp\left(-\frac{\Delta_{fg}^\psi}{\gamma_c}\right), \quad (4)$$

式中 $\Delta_{fg}^\psi = \max[|R_R(f) - R_R(g)|, |R_G(f) - R_G(g)|, |R_B(f) - R_B(g)|]$, 参数 γ_c 与参数 γ_s 类似,用来控制颜色相似度。 $P_L[\nabla_{R_f} | E_g^{(s)}(d)]$ 反映视差空间中不同层的可靠程度与纹理丰富度之间的关系,本文将其设置为

$$P_L[\nabla_{R_f} | E_g^{(s)}(d)] = f[\nabla_{R_f}, \omega(s)]. \quad (5)$$

式中 $f[\nabla_{R_f}, \omega(s)]$ 代表关于 $\omega(s)$ 和 ∇_{R_f} 的二维函数, $\omega(s)$ 是第 s 张初始视差图在代价聚合阶段使用的支撑窗口直径。总体而言,当 ∇_{R_f} 较小时, $f[\nabla_{R_f}, \omega(s)]$ 与 $\omega(s)$ 呈正相关,反之则呈负相关。 $P_L[\nabla_{D_g^{(s)}} | E_g^{(s)}(d)]$ 反映视差空间第 s 层位置 g 处视差值本身的可靠性,当 g 点所在区域视差值变化过于频繁时, g 点处反映的视差极有可能是完全错误的,因此将其设置为

$$P_L[\nabla_{D_g^{(s)}} | E_g^{(s)}(d)] = \begin{cases} 1, & \text{if } \nabla_{D_g^{(s)}} < T_{\text{threshold}} \\ 0.1, & \text{if } \nabla_{D_g^{(s)}} \geq T_{\text{threshold}} \end{cases}, \quad (6)$$

式中 $T_{\text{threshold}}$ 表示一定区域内允许的视差变化的次数。

由以上各个公式综合可以得到

$$P[E_{fg}^{(s)}, E_g^{(s)}(d) | \Delta_{fg}^\psi, \nabla_{R_f}, \nabla_{D_g^{(s)}}] \propto \exp\left(-\frac{\Delta_{f,g}}{\gamma_s}\right) \cdot \exp\left(-\frac{\Delta_{fg}^\psi}{\gamma_c}\right) \cdot f[\nabla_{R_f}, \omega(s)] \cdot P_L[\nabla_{D_g^{(s)}} | E_g^{(s)}(d)]. \quad (7)$$

对于参考图像中的某个点 f ,在其支撑窗口内有 N 个点,每一次都假设 f 与 N 个点中的一个 g 点拥有相同的视差,并计算该事件发生的似然度,如果这 N 个点中有 M 个点的视差值都为 d ,那么可以将其累加起来,作为 f 点处视差为 d 的似然度。为了表示的方便,将 f 点处视差为 d 的似然度累积值记为 $\Omega(f|d)$,其计算公式为

$$\Omega(f|d) = \sum_s \sum_{g \in N(f)} P[E_{fg}^{(s)}, E_g^{(s)}(d) | \Delta_{fg}^\psi, \nabla_{R_f}, \nabla_{D_g^{(s)}}]. \quad (8)$$

在 f 周围的各个点可能有多种视差值,将其分别进行累积,形成一个关于视差值 d 的累积直方图。本文直接挑选其中累积似然度最大的视差值作为最终的结果,即

$$\hat{d} = \operatorname{argmax}_{d \in \{d_{\min}, d_{\max}\}} \Omega(f|d). \quad (9)$$

3 实验结果

3.1 实验条件

MSO 算法的目的是对已经校正好的立体图像对进行匹配和优化。立体图像来源为 Middlebury 学院的标准测试数据集。初始视差图通过简单的 BM 方法获得。初始视差空间包括 3 张视差图,窗口半径大小分别设置为 $[N_c/150]$ 、 $[N_c/80]$ 和 $[N_c/40]$ 个像素,其中 N_c 为参考图像宽度, $[\cdot]$ 表示向下取整。之所以这么设置是因为,当窗口半径设置为 $[N_c/80]$ 时可以获得视觉上认为是最好的视差图,而当窗口半径分别设置为 $[N_c/150]$ 和 $[N_c/40]$ 时,结果明显分别带有强烈的噪声和被过度的平滑,它们分别正好包含了非常细节的信息和整体的趋势,可以为窗口半径为 $[N_c/80]$ 的结果提供深度不连续区域的位置信息和低纹理区域的深度信息。(6)式中的 $T_{\text{threshold}}$ 参数值取 $\min\{0.5, \max(\nabla_{D_g^{(s)}}) \times 0.7\}$,其中 $\min(\cdot)$ 和 $\max(\cdot)$ 分别为取最小值和

取最大值,0.5 和 0.7 为经验值。这样设置是考虑到在某些场景特别单一的情况应该适当降低阈值的大小。(8)式中邻域窗口大小为固定值 8 个像素。

本文算法由 C++ 实现,运行在一台个人电脑上,CPU 型号为英特尔酷睿 I5-3470@3.2 GHz,显卡型号为英伟达 GTX730(2 GB 显存,384 个流处理器),内存为 8 GB。

3.2 标准图像数据集的实验结果

图 3 给出了利用本文 MSO 算法对 Middlebury 学院提供的 tsukuba、venus、teddy 和 cones 立体视觉标准图像对进行测试的结果,图中其他算法的结果来自于 <http://vision.middlebury.edu/stereo/eval/> 网站,该网站对大量的算法结果进行测试和对比,本文选取了其中最为先进和影响最大的几种实时匹配算法的结果进行比较。测试表明,本文算法在保持实时计算速度的同时达到了很高的精度,和目前最先进的实时双目匹配算法相比具有非常强的竞争力。

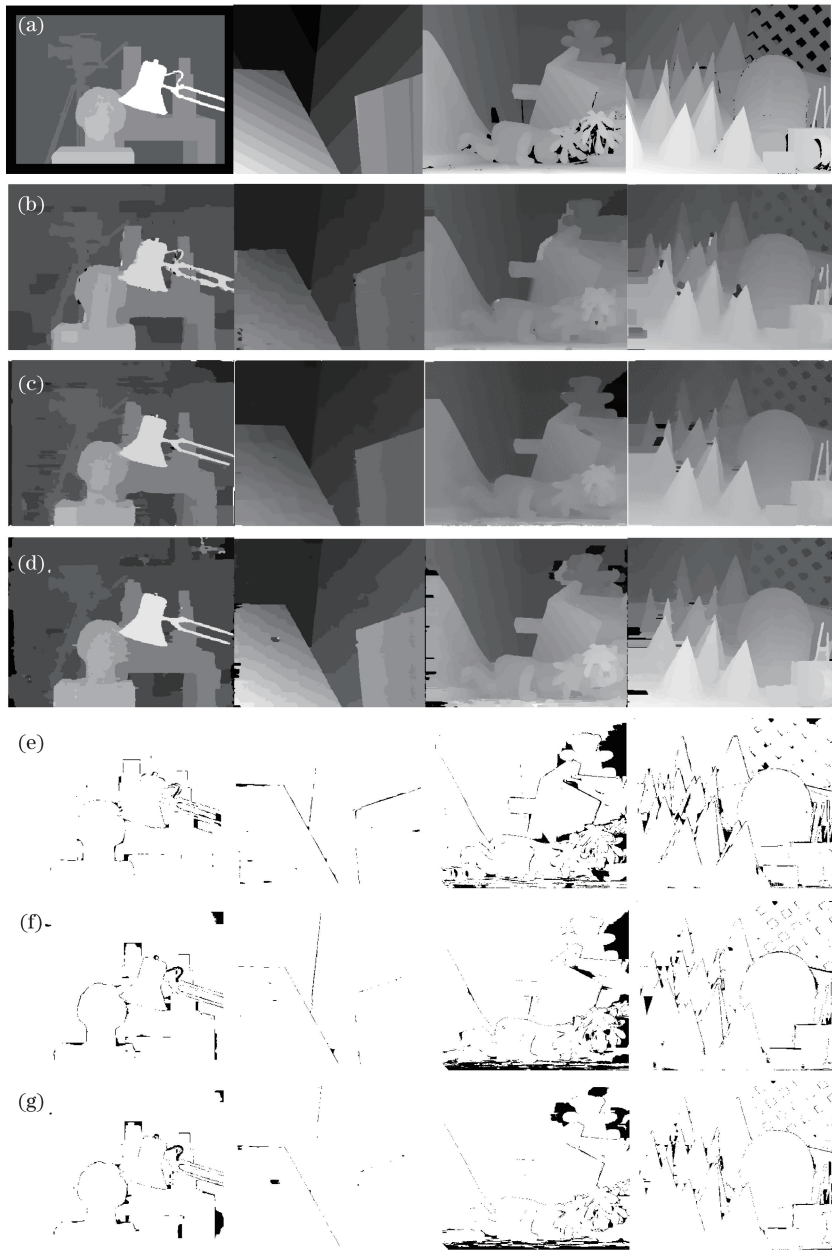


图 3 实验结果比较。(a)真实视差图;(b) RealTime BP 算法;(c) FastBilateral 算法;(d) MSO 算法;(e)~(g)对应的误匹配区域

Fig. 3 Comparison of experimental results. (a) Ground truth maps; (b) RealTime BP algorithm; (c) FastBilateral algorithm; (d) MSO algorithm; (e)-(g) corresponding mismatched regions

4 分析与讨论

4.1 精 度

表 1 是本文算法与其他实时算法的精度比较,其中各个指标的定义按照文献[30]中的说明。由表 1 的各项数据可以看出,本文算法在 venus 和 cones 图像对的表现最为突出,远超过其他各种算法,并且平均误差率也明显小于其他几项算法。产生这种现象的原因主要在于视差空间中某位置是否存在真值。正如第 3 部分所描述的,本文的出发点在于利用多幅视差图进行综合推理,从而找出使后验概率最大的视差值。以 tsukuba 和 venus 图像为例进行分析。在图 2 中,用蓝色框标示出来的部位是在所有视差空间中均无法找到正确视差的区域,而用绿色框标示的部位是在某幅视差图中有大量误匹配而在其他的视差图中消除或是大大减少了这种误匹配的区域。后一种情况是本文算法追求的核心——各幅视差图之间存在互补,因此才可以借助于综合与推理得出更好的结果。由于 tsukuba 图像边缘处存在较多的大面积无纹理区域,导致这些区域在视差空间中所有视差图都无法找到正确的视差,最终导致优化后的结果在这部分区域同样无法推理出正确的视差。这样的无纹理区域对局部算法是一个很大的挑战,而这里往往是复杂的全局算法优势最明显的区域。在 venus 图像对中,无纹理区域内部存在一定的整体变化趋势,而且在其周围有非常多的纹理丰富的区域,因此在窗口增大时能正确地检测到视差,使得视差空间中存在互补性,因而优化效果显著。在表 1 中另一个值得注意的地方是,本文算法在 disc 区域优势显著,这点在 teddy 数据集中表现明显。之所以会出现这样的结果,是因为本文算法在进行贝叶斯推理时考虑了参考图中相邻像素颜色相似性与局部平滑假设相容性的关系,相当于提取出参考图的梯度信息并对视差图进行约束,从而使得优化后的视差图在深度不连续的地方表现良好。

表 1 本文算法与其他实时算法性能比较

Table 1 Performance comparison of proposed algorithm with other real-time algorithms

Algorithm	Average	Tsukuba			Venus			Teddy			Cones		
		nocc	all	disc	nocc	all	disc	nocc	all	disc	nocc	all	disc
Ref. [20]	9.82	2.05	4.22	10.60	1.92	2.98	20.30	7.23	14.40	17.60	6.41	13.70	16.50
Ref. [23]	9.34	3.19	5.05	9.85	0.57	2.13	5.30	10.60	19.60	22.10	5.52	15.90	12.30
Ref. [19]	7.69	1.49	3.40	7.87	0.77	1.90	9.00	8.72	13.20	17.20	4.61	11.60	12.40
Ref. [31]	7.65	1.71	2.22	6.74	0.55	0.87	2.88	9.90	15.00	19.50	6.66	12.30	13.40
Ref. [32]	7.31	2.38	2.80	10.40	0.34	0.92	4.55	9.83	15.30	20.30	3.10	9.31	8.59
MSO(BM)	5.96	2.31	2.67	9.63	0.32	0.54	2.43	7.87	10.50	16.62	2.87	7.78	8.03

4.2 初始视差空间影响

为了研究初始视差空间对结果的影响,采用不同代价聚合算法得到视差图构造视差空间进行测试,同时改变多尺度寻优时的参数以适应不同的视差空间,如表 2 所示。测试结果表明,本文 MSO 算法在初始视差图较好的情况下能得到更好的结果,但是效果提升并不显著。这说明本文算法对初始视差并不敏感。产生这种现象的原因是,虽然用联合双边滤波(JBL)和引导滤波(GF)算法在设置不同支撑窗大小时产生的视差图也存在互补性,但由于 JBL 和 GF 算法具有保边平滑特性,这些算法的结果在窗口设置较大和较小时均有较低的噪声水平和较好的细节保留,因此不同窗口的结果互补性不显著。考虑到用盒滤波(BM 算法)做代价聚合要比用 JBF 和 GF 在计算速度上快非常多,本文推荐在更追求速度的情况下使用 BM 算法获取初始视差图构建视差空间。

表 2 分别使用 BM 算法和 GF 算法得到视差空间后利用本文算法进行优化的结果比较

Table 2 Performance comparison of proposed algorithm by using BM algorithm and GF algorithm to create disparity space

Algorithm	Average	Tsukuba			Venus			Teddy			Cones		
		nocc	all	disc	nocc	all	disc	nocc	all	disc	nocc	all	disc
MSO(BM)	5.96	2.31	2.67	9.63	0.32	0.54	2.43	7.87	10.50	16.62	2.87	7.78	8.03
MSO(GF)	5.64	2.34	2.61	8.85	0.31	0.54	2.70	7.53	10.33	15.22	2.58	7.50	7.16

4.3 计算复杂度

表 3 是本文算法与其他实时算法的速度比较,表中带 * 号的数据是根据相应文献中给出的数据进行换算得到的。从 MDE 数据(MDE 表示计算的百万视差数量)可以看出,本文算法在所列举的几种快速匹配算法中效率最高,这得益于本文算法的完全局部性和并行性。本文算法第一阶段是计算多幅初始视差图构建视差空间,第二阶段是对视差空间的推理和优化。第一阶段由于采用了 BM 方法,可以使评价每一像素每一视差的算法复杂度降到 $O(1)$,这一阶段算法复杂度与所使用的窗口大小无关。第二阶段对视差图中一个像素需要计算其支撑窗口内所有像素对其支撑作用的大小,之后建立累积似然度直方图并找出其中的最大值,这一步的算法复杂度为 $O(S \times n)$,其中 S 代表初始视差图数量, n 代表推理时支撑窗内像素数量。实验表明, $S=3$ 时已经能达到非常好的优化效果,而 n 的个数也可以限制在一个较小的水平(在初始视差图获取阶段已经得到了大尺度的视差信息,因此在优化阶段不需要再设置非常大的支撑窗口)。本文实验对数据集内所有数据均采用半径为 8 的支撑窗口。综合考虑,算法对每个像素每一视差的计算复杂度只有 $O(1)$ 。

表 3 本文算法与其他实时算法的运行速度比较

Table 3 Comparison of speed performance of proposed algorithm and other real-time algorithms

Algorithm	Processor	Time /ms	MDE per second
Ref. [23]	2.5 GHz CPU	8000	0.27*
Ref. [19]	3.0 GHz CPU, NVIDIAI 7900 GTX	104.9*	17.0
Ref. [32]	2.5 GHz CPU, NVIDIAI GTX460	65.5	27.2*
Ref. [20]	3.0 GHz CPU, ATI Radeon XL1800	33.6*	52.7
Ref. [31]	2.66 GHz CPU, NVIDIAI 8800 GTX	17.5	100.9
MSO(BM)	3.2 GHz CPU, NVIDIAI GTX730	15.0	106.1

5 结 论

提出了一种新颖的双目视觉匹配方法,它利用了局部算法选择不同大小支撑窗口时结果的互补性,基于贝叶斯推理对多幅带有尺度信息的视差图进行联合优化,从而得到一张高质量的视差图。基于 Middlebury 标准双目图像测试集的实验结果表明,本文算法与其他实时算法相比具有非常强的竞争力,算法的平均误差远小于其他实时算法,并在速度上保持较大的优势。

未来的重点在以下方面:1)将本文算法进一步简化,利用现场中编程门阵列(FPGA)并行实现,以便运用在机器人或其他实际应用中;2)探索利用本文算法,将用不同算法得到的带有互补信息的初始视差图进行融合,进一步提高匹配精度。

参 考 文 献

- [1] Mur-Artal R, Montiel J M M, Tardós J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.
- [2] Murray D, Little J J. Using real-time stereo vision for mobile robot navigation[J]. Autonomous Robots, 2000, 8(2): 161-171.
- [3] Sarbolandi H, Lefloch D, Kolb A. Kinect range sensing: structured-light versus time-of-flight kinect[J]. Computer Vision and Image Understanding, 2015, 139: 1-20.
- [4] Garcia F, Aouada D, Solognac T, *et al.* Real-time depth enhancement by fusion for RGB-D cameras[J]. IET Computer Vision, 2013, 7(5): 1-11.
- [5] Li L, Yu X, Zhang S, *et al.* 3D cost aggregation with multiple minimum spanning trees for stereo matching[J]. Applied Optics, 2017, 56(12): 3411-3420.
- [6] Scharstein D, Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms[J]. International Journal of Computer Vision, 2002, 47(1): 7-42.
- [7] Jin S, Cho J, Xuan D P, *et al.* FPGA design and implementation of a real-time stereo vision system[J]. IEEE Transactions on Circuits & Systems for Video Technology, 2010, 20(1): 15-26.
- [8] Yang Q. Hardware-efficient bilateral filtering for stereo matching[J]. IEEE Transactions on Pattern Analysis &

- Machine Intelligence, 2014, 36(5): 1026-1032.
- [9] Zhu Shiping, Li Zheng. 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.
- [10] Hosni A, Rhemann C, Bleyer M, *et al.* Fast cost-volume filtering for visual correspondence and beyond[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 35(2): 504-511.
- [11] Xu S, Zhang F, He X, *et al.* PM-PM: PatchMatch with Potts model for object segmentation and stereo matching[J]. IEEE Transactions on Image Processing, 2015, 24(7): 2182-2196.
- [12] Sitnik R, Puech W, Peng Y, *et al.* Stereo matching with space-constrained cost aggregation and segmentation-based disparity refinement[C]. SPIE, 2015, 9393: 939309.
- [13] Mozerov M G, Van De Weijer J. Accurate stereo matching by two-step energy minimization[J]. IEEE Transactions on Image Processing, 2015, 24(3): 1153-1163.
- [14] Yedidia J S, Freeman W T, Weiss Y. Understanding belief propagation and its generalizations[J]. Exploring Artificial Intelligence in the New Millenium, 2002, 54(1): 276-286.
- [15] Boykov Y, Veksler O, Zabih R. Fast approximate energy minimization via graph cuts[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2001, 23(11): 1222-1239.
- [16] Zhu Shiping, Yan Li'na, Li Zheng. Stereo matching algorithm based on improved Census transform and dynamic programming[J]. Acta Optica Sinica, 2016, 36(4): 0415001.
祝世平, 闫利那, 李政. 基于改进 Census 变换和动态规划的立体匹配算法[J]. 光学学报, 2016, 36(4): 0415001.
- [17] Hirschm H. Stereo processing by semiglobal matching and mutual information[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2008, 30(2): 328-341.
- [18] Ernst I, Hirschmiller H. Mutual information based semi-global stereo matching on the GPU[C]. International Symposium on Visual Computin, 2008: 228-239.
- [19] Yang Q X, Wang L, Yang R G, *et al.* Real-time global stereo matching using hierarchical belief propagation[C]. British Machine Vision Conference, 2006, 6: 989-998.
- [20] Wang L, Liao M, Gong M, *et al.* High-quality real-time stereo using adaptive cost aggregation and dynamic programming[C]. International Symposium on 3D Data Processing, Visualization, and Transmission, 2006: 798-805.
- [21] Mattoccia S, Giardino S, Gambini A. Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering[C]. Asian Conference on Computer Vision, 2009: 371-380.
- [22] Ma Z, He K, Wei Y, *et al.* Constant time weighted median filtering for stereo matching and beyond[C]. IEEE International Conference on Computer Vision, 2013: 49-56.
- [23] Mattoccia S. A locally global approach to stereo correspondence[C]. IEEE International Conference on Computer Vision Workshops, 2010: 1763-1770.
- [24] Felzenszwalb P F, Huttenlocher D P. Efficient belief propagation for early vision[J]. International Journal of Computer Vision, 2006, 70(1): 41-54.
- [25] Liang C, Wang L, Liu H. Stereo matching with cross-based region, hierarchical belief propagation and occlusion handling[C]. 2011 IEEE International Conference on Mechatronics and Automation, 2011: 1999-2003.
- [26] Zhang K, Fang Y, Min D, *et al.* Cross-scale cost aggregation for stereo matching[C]. 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014: 1590-1597.
- [27] Li X, Song Z. Optimization on stereo correspondence based on local feature algorithm[C]. 2017 2nd International Conference on Image, Vision and Computing, 2017: 113-117.
- [28] Yang H, Yu M, Zhang S. Wide baseline stereo matching based on scale invariant feature transformation with hybrid geometric constraints[J]. IET Computer Vision, 2014, 8(6): 611-619.
- [29] Scharstein D, Szeliski R. High-accuracy stereo depth maps using structured light[C]. 2003 IEEE Conference on Computer Vision and Pattern Recognition, 2003: 195-202.
- [30] Scharstein D, Szeliski R, Hirschmüller H. Middlebury stereo vision page[EB/OL]. (2015-12-16)[2017-06-30]. <http://vision.middlebury.edu/stereo/>.
- [31] Zhang K, Lu J, Lafruit G, *et al.* Real-time accurate stereo with bitwise fast voting on CUDA[C]. IEEE International Conference on Computer Vision Workshops, 2010: 794-800.
- [32] Mattoccia S, Viti M, Ries F. Near real-time fast bilateral stereo on the GPU[C]. Computer Vision and Pattern Recognition Workshops, 2011: 136-143.