

基于区域增长的稠密立体匹配

刘 杰, 张建勋, 代 煜

(南开大学机器人与信息自动化研究所, 天津 300071)

摘 要: 针对稠密立体匹配中支持窗口的尺寸、形状和深度相似像素难以选择, 以及物体边界和遮挡区域难以处理的问题, 提出一种基于区域增长的局部立体匹配算法. 首先应用区域增长的方法, 通过颜色相似性和连通性约束动态地获得完全自适应的支持窗口. 然后运用不同的策略为支持窗口内像素和搜索窗口内其他像素分配权值, 减弱非深度相似像素对支持窗口的影响, 并摒弃视差范围内不符合形状相似标准的支持窗口. 最后, 计算匹配代价. 应用 WTA (赢者全取) 策略获得初始视差, 统计支持窗口内符合置信度条件的像素的视差频率, 以出现频率最高的视差作为最优视差. 通过对支持区域获取、代价聚合和视差搜索步骤的分析和优化, 降低算法复杂度. 通过 Middlebury 平台的检验, 表明算法具有良好的性能.

关键词: 局部立体匹配; 区域增长; 形状相似性; 支持窗口; 视差精调

中图分类号: TP391.4

文献标识码: A

文章编号: 1002-0446(2017)-02-0182-07

Dense Stereo Matching Based on Region Growing

LIU Jie, ZHANG Jianxun, DAI Yu

(Institute of Robotics & Automatic Information System, Nankai University, Tianjin 300071, China)

Abstract: For dense stereo matching, a local stereo matching algorithm based on region growing is proposed to solve the difficulties in choosing the size and shape of support windows as well as the pixels that have similar depth, and the difficulties in dealing with the boundaries of objects as well as the occlusion areas. Firstly, color similarity and connectivity constraint are adopted to obtain the adaptive support window dynamically by means of a method based on region growing. Then, the weights of pixels in support window or search window are assigned by different strategies, to reduce the effect of the pixels with different depths on the support window, and abandon the support windows that don't conform to the shape similarity rule. Finally, the matching cost is computed. The initial disparity is obtained by means of WTA (winner takes all) strategy. For the support windows, the frequencies of disparities of the pixels in accord with the confidence level standard are computed, in which the disparity with the highest frequency is taken as the optimal disparity. Moreover, the algorithm complexity is reduced through analyzing and optimizing the processes of support region computation, cost aggregation, and disparity search. The proposed algorithm is tested on Middlebury platform, and the result shows its excellent performance.

Keywords: local stereo matching; region growing; shape similarity; support window; disparity refinement

1 引言 (Introduction)

立体匹配是多目视觉中一个重要的研究课题. 根据优化方式的不同, 基本可以把立体匹配划分为全局立体匹配和局部立体匹配两类. 现行的全局匹配方案如置信度传播^[1]、图割^[2]等虽然有较高的匹配精度, 但由于算法包含的参数较多, 其复杂度过高, 往往需要分布式系统和专用计算设备的支持, 难以满足实时性要求. 在局部立体匹配方案中, 由于图像中可能存在遮挡、畸变、噪声等因素的影响, 计算某一像素的深度时, 往往需要邻域像素的支持. 而这些提供支持支持的邻域像素, 必须具备深度

相似性. 因此, 支持窗口的形状、尺寸和深度相似像素的选择, 以及对物体边界 (一般是深度发生变化的区域) 和遮挡点的处理, 一直是局部立体匹配方案中的难题.

支持窗口的选择通常取决于图像中纹理和视差的局部变化情况. 一般来说, 较小的窗口可以避免过平滑, 然而包含信息较少, 可能导致弱纹理区域的误匹配, 并且容易受到噪声影响; 较大的窗口对弱纹理区域匹配较好, 但会导致深度不连续区域的误匹配, 容易引起“前景膨化”.

为了解决支持窗口的选择问题, 文 [3] 提出了

多窗口法, 算法基本思想是在几个预设定的不同大小和形状的窗口中, 选择一个性能最好的予以保留, 但是这种方法不但计算量较大, 且因为预设定的窗口无论在大小和形状上都是有限的, 对匹配精度的改善很有限. 为了使支持窗口的选择更加自由, 文 [4] 提出了自适应窗口方法, 通过灰度的局部变化情况和预估计的视差值自适应地决定窗口的大小和形状, 虽然在一定程度上提高了匹配精度, 但是窗口形状局限于矩形, 且时间复杂度很高. 为了进一步提升匹配精度, 文 [5] 提出了自适应权值法, 通过对窗口内每个像素颜色和距离的衡量, 为其分配相应的权值, 算法旨在弱化深度非相似像素的影响, 而不是尝试去除这些像素的影响, 所以并没有真正解决支持窗口的选择问题. 文 [6] 在匹配代价上做了较深入的研究, 提出了一种将多种相似性测度相结合的匹配方案, 匹配精度提升较为明显.

基于分割的方法^[7-9]在一定程度上解决了支持窗口的选择问题, 但是对于遮挡点和物体边界的处理较差. 基于区域增长的方法^[10]在某种程度上也可以看成是基于分割的, **不同的是区域增长的方法自始至终具备连通性约束**, 且与传统的基于分割的方法相比, 在支持窗口的选择上更具备多样性. 通过文 [11-12] 等的改进和提升, 可见基于区域增长的方法对提升匹配精度是有效的. 但是传统的基于区域增长的方法通常会先确定一批可信度较高的种子点, 之后通过一定的规则把这批像素的匹配关系传播至其余可信度较低的像素上, 即可信度较低的像素匹配精度依赖于原始种子点. 这样的匹配方案过于依赖种子点的匹配精度, 且一旦中途点匹配出现误差, 由此传播的其他像素也会随之出现误差, 这种误差累积的危害性对整体匹配精度影响较大, 而且这种误差积累难以从根本上予以避免, 并且在遮挡点聚集区域和大面积弱纹理区域, 往往匹配精度较差.

近年来, 一些最新的成果尝试从滤波和特征融合的角度来解决立体匹配问题. 文 [13] 基于最小生成树, 将权重中值滤波等价于代价聚合, 使得中心像素的邻域包含整幅图像的所有像素, 虽然得到的权值更为精确, 但是是以相当大的时间复杂度作为代价的. 文 [14] 把传统的代价聚合改进为扩展的二值加权聚合, 虽然使得聚合后的匹配代价更为准确, 但是一方面复杂度较高, 另一方面面对深度不连续区域的匹配精度并没有随之提升. 文 [15] 基于分割得到了中心像素的支持窗, 这样的策略与本文

基于区域增长的方法有一定类似之处, 但论文随后对不同的特征进行了融合, 虽然匹配精度有一定提升, 但这种特征融合的策略并不具备普遍性, 很难说明这样几种特征的融合适用于所有的立体图.

基于以上讨论, 本文提出一种新的基于区域增长的立体匹配算法. 算法分为 3 步: (1) 利用颜色相似性和连通性约束, 通过区域增长的方法来获取支持区域, 使得获取的支持窗口内的像素尽可能具备深度相似性. (2) 运用不同的策略为支持窗口内像素和搜索窗口内其他像素分配权值, 进一步减弱非严格深度相似像素对支持窗口的影响, 通过形状相似性准则摒弃视差范围内不符合形状相似标准的支持窗口. (3) 计算匹配代价, 并通过对支持区域获取、代价聚合和视差搜索步骤的分析和优化, 降低算法复杂度, 应用 WTA 策略获得初始视差, 统计支持窗口内符合置信度条件的像素的视差, 以出现频率最高的视差作为最优视差.

2 区域增长立体匹配 (Stereo matching based on region growing)

2.1 区域增长策略

为了消除匹配歧义, 获得图片中某个像素的准确视差值, 局部匹配方案一般会以每个像素为中心建立一个支持区域. 通过对立体图中该像素支持区域内像素的计算, 以该区域的匹配代价作为该像素的匹配代价, 最终选择最小匹配代价对应像素的视差作为最优视差. 但是, 并不是相邻的所有像素均可以作为支持像素, **根据立体匹配中常用的前向平行假设**, **可以得知支持区域内的像素需具备深度相似性**. 而在实际立体图中, 每个像素的支持区域千差万别, 因此如何选择最优的支持区域长期以来都是立体匹配中的一个难题, 本节应用区域增长的方法, 试图求取基于中心像素本身特性得到的自适应支持区域.

在立体匹配中, **一个基本的假设是: 深度不连续发生在颜色边缘, 而通常来说颜色边缘也是物体边界区域**. 从这个假设出发, 基于连续性约束, 首先选定初始种子点 S_0 , **即种子点属于第 0 代种子**, 之后通过一定的规律寻找连通像素中的下一代种子, 然后以下一代种子作为新一代种子点, **进而寻找这些新的种子像素的支持像素, 而后循环寻找支持像素, 直至最后满足循环停止条件**. 这样以种子像素本身特性寻找支持像素, 一步步扩展支持区域, 以期得到基本符合中心像素视差特性的支持区域, 为之后的代价聚合做好准备. 每一代种子的寻

找依据为颜色相似性, 表示为

$$S(p, q) = \sum_{j \in \{r, g, b\}} \sqrt{(d_j^p - d_j^q)^2} \quad (1)$$

其中, d_j^p 和 d_j^q 分别是像素 p 和 q 在颜色通道 j 上的数值. 若 $S(p, q)$ 小于预定阈值 T , 则判定像素 q 属于第 $i+1$ 代种子.

具体的区域增长策略如下:

- 1) 选择初始种子点 S_0 , 把 S_0 作为第 0 代种子.
- 2) 寻找第 1 代种子. 用式 (1) 对 S_0 的连通像素 C_0^i 进行判定, 符合条件的像素放入区域 $r[1]$ 中.
- 3) 取 $i = 1$, 从 $r[i]$ 中依次取出每个像素 S_i 作为第 i 代种子, 在其连通像素 C_i 中, 开始计算符合条件的像素放入区域 $r[i+1]$ 中. 需要注意的是, 在种子点 S_i 的自适应支持窗口计算中, 如果像素 S_i 曾经被公式判定过该像素属于支持区域, 则在后续的计算中, 不再计算该像素. i 自加 1.
- 4) 重复步骤 3), 直到达到预定的代数 g 或公式判定区域 $r[i+1]$ 中有效像素数目为 0.

为了更好地展示区域增长的过程, 图 1 截取了 Middlebury 测试平台^[16]上 cones 图像的左上角, 以 $(0, 0)$ 作为初始种子点开始计算支持窗口.

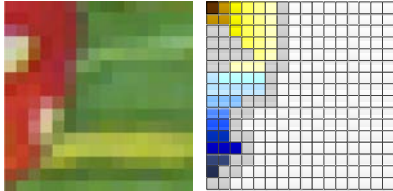


图 1 支持区域增长过程

Fig.1 The growing process of support region

左上角像素是初始种子像素, 每一代种子像素颜色依次由暖变冷, 其中浅灰色点为连通像素中经过判定不属于种子点的像素.

下面选取一幅深度相似区域比较复杂的图, 进一步展示区域增长方案得到的支持区域, 在图 2 上任意选取叶柄上的一个像素作为初始种子像素, 开始计算其支持区域.



图 2 由叶柄上某像素得到的支持窗口

Fig.2 The support region of a pixel in petiole

可以看出, 当设定一个合适的阈值 T 时, 基于区域增长方案可以很高效地找出种子像素的支持

区域. 无论深度相似区域的形状如何复杂, 均不影响最终的效果. 支持区域的形状和尺寸在区域增长的过程中都是充分自适应的, 只依赖于颜色相似性和连通性, 且最大程度上使得支持窗口内像素具备深度相似性, 极大地降低了立体图匹配中可能产生的匹配歧义.

值得说明的是, 本文的算法与基于分割的方法^[7-8]和自适应窗口方法^[4]的区别在于: 自适应区域增长策略得到的支持窗口自始至终是基于连通性的, 即与种子像素不具备连通性的像素, 即使距离接近、颜色相似, 依然不能进入支持区域. 这样得到的支持窗口在匹配代价计算中更加稳定, 并且在匹配过程中不容易产生歧义.

为了在支持区域内获得较为严格的深度相似, 一般把阈值 T 设置得较小, 但是这样的选择会导致一些像素的支持窗口很小. 针对这样的像素, 当其支持区域的像素数小于一定数值 T_n 时, 本文选择采用最大尺寸为 M 的自适应窗口算法来获取支持区域.

2.2 物体边界和遮挡点

通常来说, 图片中物体的边缘往往是深度发生变化的地方, 但是实际立体图中物体深度结构复杂且相邻物体可能发生重叠而深度截然不同, 因此深度不连续像素的处理成为立体匹配中较难处理的像素. 当确定待匹配像素 p 时, 通过上述方案可以得到像素 p 的支持区域 R , 候选匹配像素 \hat{p} 的支持区域 \hat{R} . 若像素 p 在物体 O 上, p_i 邻近于物体 O 的边界且已经不属于物体 O , 那么一般来说, 通过颜色相似性和连通性约束得到的 p 和 p_i 的支持区域是截然不同的. 通过一定的规则对区域 R 和 \hat{R} 进行比较, 可以有效地处理深度不连续的像素.

针对这类像素, 本节定义了形状相似性来表达这种支持区域之间的区别. 由于处于不同深度区域的像素得到的支持区域通常差别很大, 因此本文定义的形状相似性主要通过支持区域内像素的数目以及平均每行像素的数目来衡量. 若立体图的支持区域不满足这样的形状相似性判定, 则视为非候选匹配像素 (这其中包括了干扰性很强的深度区域边缘像素). 支持窗口的形状相似性比较在具体实现中分为 2 层: 首先比较支持区域内像素的总数目, 然后比较行数目. 这种由粗到细的比较也符合人类的基本视觉判断规律. 通过这样的相似性判定, 基本可以满足之后权值分配和代价聚合步骤实施的要求.

由于遮挡、光照、拍摄角度等因素的影响, 候

选匹配像素 \hat{p} 的支持区域 \hat{R} 和待匹配像素 p 的支持区域 R 的形状很难完全相同, 因此这里定义一个窗口相同的阈值, 如果 2 个像素支持窗口的相似性超过这个阈值 T_s , 就认为 2 个窗口形状相同. 由上所述, 形状相似性定义为

$$f_{sh}(p, \hat{p}) = f(N_p, N_{\hat{p}}) \cdot f(N_p^{avg}, N_{\hat{p}}^{avg}) \quad (2)$$

其中, f 函数为

$$f(a, b) = \begin{cases} 1, & \text{if } a(1 - \alpha) < b < a(1 + \alpha) \\ 0, & \text{else} \end{cases} \quad (3)$$

其中, N_p 和 $N_{\hat{p}}$ 是像素 p 和 \hat{p} 的支持区域内有效像素的数量, N_p^{avg} 和 $N_{\hat{p}}^{avg}$ 是支持区域内有效像素平均每行的数量, $1 \pm \alpha$ 是阈值 T_s . 因为支持区域是逐代增长的, 因此支持区域内像素的总数目和平均每行数目, 可以在支持窗口计算的过程中同步获得, 几乎不增加额外的计算消耗.

当待匹配像素和候选匹配像素位于同一深度区域时, 形状相似性判定不起任何作用. 但是当像素位于物体边缘或者属于遮挡区域时, 形状相似性可以很容易地判别出来. 利用这一特性, 可以很高效地处理物体边界像素和遮挡像素, 客观上也降低了算法的复杂度.

图 3 展示了 2 个相邻像素, 其中一个处于物体边界, 另一个处于物体边界外时, 得到的支持区域.

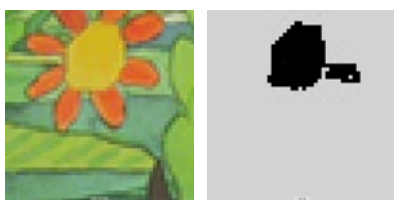


图 3 2 个物体边界像素的支持窗口

Fig.3 The support windows of 2 pixels in the object boundary

若是视其中一个像素为候选匹配像素, 采用传统算法很难剔除这样 2 个相邻但隶属于完全不同的深度区域的强干扰像素. 而利用基于区域增长的方法计算支持区域, 加上形状相似性的判定, 可以看出, 2 个支持区域无论在形状和尺寸上都存在明显区别. 通过简单计算, 区域 a 内总像素数目为 202, 平均每行像素为 11.9; 区域 b 内总像素数目为 51, 平均每行像素为 7.3. 可见, 形状相似性完全可以表达处于物体边界时 2 个像素的支持窗口间的区别.

2.3 权值分配和代价聚合

自适应窗口的方法假设窗口内像素的深度值是一样的, 但是事实上并非如此, 由于阈值 T 的存在, 窗口内像素并非严格的深度相似. 经过对立体图的分析可知, 支持区域内像素是缓慢变化的, 且变化的幅度很小. 因此为了弱化非严格深度相似像素对代价聚合的影响, 本文为自适应支持窗口内的每个像素分配了相应的权值.

同时为了防止由于 T 设置得过小, 从而引起支持区域与实际深度相似区域差别较大, 同样为搜索窗口内不属于支持区域的像素分配权值.

权值的分配策略参照文 [5] 的方案, 但是对支持区域内的像素只考虑颜色相似性, 对搜索窗口内不属于支持区域的像素则同时考虑距离邻近性和颜色相似性. 这样的权值分配策略使得支持区域内像素的权值较大, 且变化较小; 随着距离邻近性的加入, 搜索窗口内不属于支持区域的像素权值较小, 变化却较大. 通过这样的权值分配策略, 既保证了支持区域的相对严格深度相似性, 也保证了搜索区域内其他像素较弱但不容忽视的支持作用. 权值的计算主要通过颜色相似性和距离邻近性来判定, 表示为

$$\omega(p, q) = k \cdot f_s(\Delta C_{pq}) \cdot f_p(\Delta g_{pq}) \quad (4)$$

其中, p 为参考图中的中心像素, q 为中心像素经过判定的支持像素, $f_s(\Delta C_{pq})$ 和 $f_p(\Delta g_{pq})$ 分别表示颜色相似性和距离邻近性的聚类强度, ΔC_{pq} 和 Δg_{pq} 分别表示中心像素 p 和支持像素 q 的颜色相似程度和距离邻近程度, k 为权值的控制系数. 值得注意的是, $f_p(\Delta g_{pq})$ 在像素 p 属于支持区域的时候值恒为 1. 由于支持区域的生长也基于颜色相似性, 因此在支持区域中颜色相似性的计算可以重复利用式 (1) 中的颜色相似性 $S(p, q)$ 和预定阈值 T , 定义为

$$f_s(\Delta C_{p_n q_{n+1}}) = \prod_{0 < n < i} \left(1 - \frac{S(p_n, q_{n+1})}{T} \right) \quad (5)$$

其中下标 n 表示属于第几代种子像素, i 为种子像素的最大代数.

代价聚合基于之前得到的支持像素的权重和立体图像像素间的原始匹配代价进行计算, 表示为

$$E(p, \hat{p}_d) = \frac{\sum_{q \in N_p} \sum_{\hat{q}_d \in N_{\hat{p}_d}} \omega(p, q) \omega(\hat{p}_d, \hat{q}_d) e_o(q, \hat{q}_d)}{\sum_{q \in N_p} \sum_{\hat{q}_d \in N_{\hat{p}_d}} \omega(p, q) \omega(\hat{p}_d, \hat{q}_d)} \quad (6)$$

其中, p 、 q 、 \hat{p}_d 、 \hat{q}_d 分别表示参考图中心像素、参考图支持像素、匹配图在视差为 d 时的中心像素和

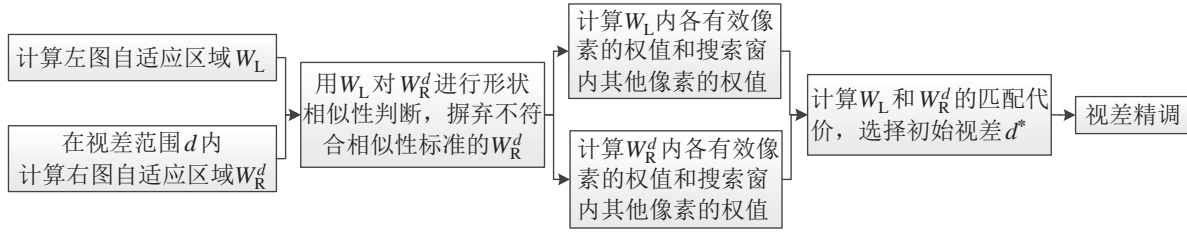


图 4 本文算法流程图

Fig.4 Flow chart of the proposed algorithm

支持像素, N_p 和 N_q 分别表示参考图和匹配图中心像素的支持区域, $e_o(q, \hat{q}_d)$ 是像素之间的原始匹配代价.

2.4 视差选择及精调

在完成像素非相似性计算以后, 采用 WTA 策略来选择初始视差. 定义初始视差为

$$d_p = \arg \min_{d \in S_d} E(p, \hat{p}_d) \quad (7)$$

在支持窗口内, 如果某像素与初始种子像素颜色相似性大于置信度 T_c , 则认为该像素和种子像素来自于相同的深度. 统计这些像素点视差 d_s 出现的频率 $\eta(d_s)$, 然后把出现频率最高的视差当成最优视差. 该最优视差定义为

$$d_p^* = \arg \max_{d_s} \eta(d_s) \quad (8)$$

至此, 整个算法的流程图如图 4 所示.

2.5 算法分析和优化

经过分析得知, 在支持区域计算和代价聚合过程中, 需要为每个像素计算支持区域和匹配代价, 这样的计算消耗极为巨大, 算法复杂度为 $O(HWDN)$, 其中 H 、 W 为图片高度和宽度, D 为视差搜索范围, N 为平均支持窗口的大小.

经过分析, 算法的计算主要消耗在支持区域的获取、代价聚合以及视差搜索上. 针对这些步骤的计算消耗, 下面逐一分析优化, 以降低算法的时间复杂度, 从而保证算法的实时性.

(1) 支持区域获取. 一般每个像素只属于唯一的一个支持区域, 如果像素 p 的支持区域中包含像素 p_i , 那么就可以不再计算 p_i 的支持区域, 从而使得支持区域针对一部分像素是共享的. 假设图片中有 S 个支持窗口, 平均每个窗口的尺寸为 $a \times b$, 那么算法的时间复杂度降低为 $O(HWDN_i)$, 其中 $N_i = N/(a \times b)$.

(2) 代价聚合. 在颜色相似性的计算中, 像素 p 和 q 与 q 和 p 的颜色相似性是相同的. 距离邻近性是不用在支持区域内逐次计算的, 如果设定了代

数 g , 相当于设定了种子点区域增长的边界, 像素的距离邻近性可以进行一次计算而重复利用.

(3) 缩小视差搜索范围. 在视差范围内计算匹配代价时, 像素 p_i 的匹配代价 C_d 可以重复利用之前一个像素除 $C_{d_{\max}}$ 外的所有匹配代价. 在视差范围内进行代价聚合时, 先对候选匹配像素 p_i 与待匹配像素 p 的支持窗口进行形状相似性判定, 若不符合, 则放弃像素 p_i . 根据之前形状相似性的定义, 可知支持窗口形状不相似的像素基本不会是最佳匹配像素.

3 实验结果和分析 (Experiment results and analysis)

为了检验算法性能, 在 Intel core i5 3.4 GHz, 8 G 内存平台下用 VS2013 结合 OpenCV2.4.9 编程实现了本文算法, 使用 Middlebury 平台^[16]提供的图片进行了测试. 算法的经验参数设置情况为: $T = 13.5$, $g = 50$, $T_n = 9$, $M = 10$, $\alpha = 0.7$, $T_c = 0.8$. 具体实验结果如图 5 所示.

从图中可以看出, 本文算法在低纹理区域、物体边界区域和深度不连续区域得到的视差图均保持了较高的准确性. 在局部立体匹配算法中, 与目前先进的 LM3C^[17]、FastBilateral^[18]、AsswGrad^[19]等算法相比, 本文得到的视差图是更为准确的. 与全局匹配算法 GlobalGCP^[20]相比较, 本文算法也取得了不错的结果, 在低纹理区域和深度不连续区域得到的结果也更胜一筹, 但是相较于全局匹配方案大幅降低了时间复杂度, 更符合实时性要求.

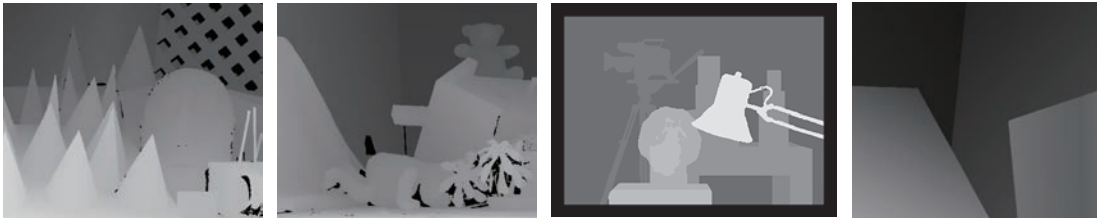
表 1 展示了上述算法在 Middlebury 测试平台上的误匹配率. 表格中 “n-occ”、“all”、“disc”、“AvgPBM” 的意思分别是非遮挡区域误匹配像素比、总体误匹配像素比和深度不连续区域误匹配像素比、平均误差. 具体到每幅测试图, 可见由于 Teddy 图中弱纹理区域 (如淡蓝色背景板)、重复纹理 (如砖墙部分) 和遮挡点 (真实视差图黑色部分) 相对较多, 且深度不连续区域变化复杂, 因此在 4 幅测试图中匹配精度最低; Cones 图的画面深

度结构复杂, 各个圆锥使得图中深度不连续区域较多, 且近背景处遮挡区域较大, 因此匹配精度也较低; Tsukuba 图中, 画面远近层次较为分明, 但是每个深度层次的区域构成最为复杂, 这也造成了支持区域获取困难, 且深度不连续区域在这部分分布较多, 拉低了立体图的整体匹配精度; Venus 测试

图结构较为简单, 画面纹理较多且不是重复模式, 因此匹配精度最高. 但是从整体上看, 本文算法依然表现良好, 分析表中的数据可知, 本文算法无论在平均误差还是深度不连续区域相较于其他算法均具备一定的优势, 在深度不连续区域的性能优势相对比较明显.



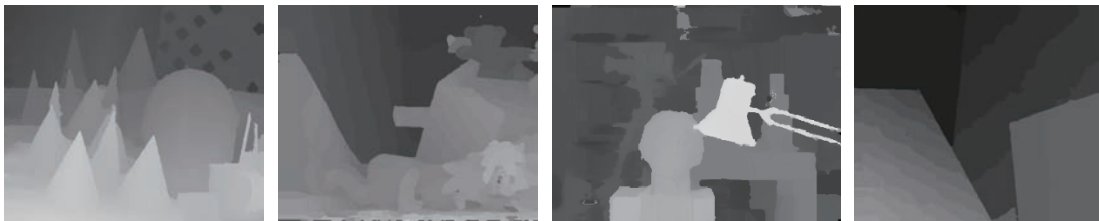
(A) 原始彩色左图像



(B) 真实视差图像



(C) 本文算法视差图像



(D) 视差精调后视差图

图 5 本文算法视差图

Fig.5 The disparity maps of the proposed algorithm

表 1 不同算法误匹配率比较
Tab.1 The AvgPBM of different algorithms

算法	Tsukuba			Venus			Teddy			Cones			Avg PBM
	n-occ	all	disc	n-occ	all	disc	n-occ	all	disc	n-occ	all	disc	
本文算法	2.07	2.32	7.39	0.13	0.38	1.12	5.90	10.1	14.6	2.10	7.40	6.10	5.09
LM3C	2.10	2.44	8.01	0.12	0.39	1.23	5.46	10.9	14.9	2.12	7.59	6.14	5.12
FastBilateral	2.38	2.80	10.4	0.34	0.92	4.55	9.83	15.3	20.3	3.10	9.31	8.59	7.31
AsswGrad	1.57	2.00	7.32	0.89	1.00	3.18	7.20	12.4	16.1	3.68	9.18	8.62	6.10
GlobalGCP	0.87	2.54	4.69	0.46	0.53	2.22	6.44	11.5	16.2	3.59	9.49	8.9	5.60

4 结论 (Conclusion)

本文提出了一个基于区域增长的局部立体匹配算法。本文算法中支持区域的计算是基于区域生长的, 因此可以获得任意形状和大小的支持窗口, 且最大程度上保留了窗口内像素的深度相似性, 真正得到了和实际视差相对应的支持窗口, 从而有效解决了由于窗口选择带来的匹配精度问题。通过为支持窗口内像素和搜索窗口内其他像素分配相应权值, 减弱了非严格深度相似像素对支持窗口以及匹配精度的影响。通过对 2 幅图像支持窗口进行多层的形状相似性判断, 不仅降低了遮挡和物体边界像素对视差估计的影响, 还能有效地估计视差搜索范围, 降低了算法的时间复杂度。

实验结果表明, 本文的算法具备良好的匹配精度。

参考文献 (References)

- [1] Banno A, Ikeuchi K. Disparity map refinement and 3D surface smoothing via directed anisotropic diffusion[J]. *Computer Vision and Image Understanding*, 2011, 115(5): 611-619.
- [2] Saygili G, van der Maaten L, Hendriks E A. Feature based stereo matching using graph cuts[C]//*Proceedings of ASCI-IPA-SIKS Tracks*. 2011: 1-6.
- [3] Fusiello A, Roberto V, Trucco E. Efficient stereo with multiple windowing[C]//*IEEE Conference on Computer Vision and Pattern Recognition*. Piscataway, USA: IEEE, 1997: 858-863.
- [4] Kanade T, Okutomi M. A stereo matching algorithm with an adaptive window: Theory and experiment[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994, 16(9): 920-932.
- [5] 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.
- [6] 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.
Lu D, Lin X. A local stereo matching algorithm based on the combination of multiple similarity measures[J]. *Robot*, 2016, 38(1): 1-7.
- [7] Gerrits M, Bekaert P. Local stereo matching with segmentation-based outlier rejection[C]//*3rd Canadian Conference on Computer and Robot Vision*. Piscataway, USA: IEEE, 2006.
- [8] Tombari F, Mattoccia S, Stefano L D. Segmentation-based adaptive support for accurate stereo correspondence[C]//*Advances in Image and Video Technology, Second Pacific Rim Symposium*. Berlin, Germany: Springer, 2007: 427-438.
- [9] 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.
- [10] 唐丽, 吴成柯, 刘侍刚, 等. 基于区域增长的立体像对稠密匹配算法[J]. *计算机学报*, 2004, 27(7): 936-943.
Tang L, Wu C K, Liu S G, et al. Image dense stereo matching by technique of region growing[J]. *Chinese Journal of Computers*, 2004, 27(7): 936-943.
- [11] 胡汉平, 朱明. 基于种子点传播的快速立体匹配[J]. *光学精密工程*, 2015, 23(3): 887-894.
Hu H P, Zhu M. Fast stereo matching based on seed pixel propagation[J]. *Optics and Precision Engineering*, 2015, 23(3): 887-894.
- [12] Wang P, Wu F. A local stereo matching algorithm based on region growing[C]//*Advances on Digital Television and Wireless Multimedia Communications*. Berlin, Germany: Springer, 2012: 459-464.
- [13] Yang Q X. Stereo matching using tree filtering[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, 37(4): 834-846.
- [14] Kim S, Kang S J, Kim Y H. Real-time stereo matching using extended binary weighted aggregation[J]. *Digital Signal Processing*, 2016, 53(C): 51-61.
- [15] Shi H, Zhu H, Wang J, et al. Segment-based adaptive window and multi-feature fusion for stereo matching[J]. *Journal of Algorithms & Computational Technology*, 2016, 10(1): 3-11.
- [16] Scharstein D, Szeliski R. Middlebury stereo vision page [DB/OL]. [2016-10-15]. <http://vision.middlebury.edu/stereo/>.
- [17] Lee Z, Juang J, Nguyen T Q. Local disparity estimation with three-moded cross census and advanced support weight[J]. *IEEE Transactions on Multimedia*, 2013, 15(8): 1855-1864.
- [18] Mattoccia S, Giardino S, Gambini A. Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering[C]//*9th Asian Conference on Computer Vision*. Berlin, Germany: Springer, 2009: 371-380.
- [19] 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): 115-123.
Zhu S P, Li Z. A stereo matching algorithm using improved gradient and adaptive window[J]. *Acta Optica Sinica*, 2015, 35(1): 115-123.
- [20] Wang L, Yang R G. Global stereo matching leveraged by sparse ground control points[C]//*IEEE Conference on Computer Vision and Pattern Recognition*. Piscataway, USA: IEEE, 2011: 3033-3040.

作者简介:

刘 杰 (1990-), 男, 博士生. 研究领域: 计算机视觉, 立体匹配, 机器学习.

张建勋 (1961-), 男, 博士, 教授. 研究领域: 医疗图像处理, 智能机器人控制, 计算机网络与信息系统.

代 煜 (1981-), 男, 博士, 副教授. 研究领域: 图像处理, 医疗机器人技术, 微弱信号处理.