

中图法分类号: TP391.4 文献标志码: A 文章编号: 1006-8961(2011)09-1664-06

论文索引信息: 袁满, 刘智, 王番, 胡涛. 改进的局部加权窗口匹配代价算法 [J]. 中国图象图形学报, 2011, 16(9): 1664-1669

改进的局部加权窗口匹配代价算法

袁满^{1),2)}, 刘智¹⁾, 王番¹⁾, 胡涛²⁾

¹⁾(信息工程大学测绘学院, 郑州 450052) ²⁾(78155 部队, 成都 610036)

摘要:稠密匹配可以看做匹配代价最小化的过程, 而匹配代价的计算是各种稠密匹配算法的必要步骤。分析取得良好效果的测地线距离局部加权窗口匹配算法, 针对测地线权值计算的核心部分进行改进, 提出一种环式的测地线权值计算方法, 并详细介绍改进后的权值计算方法应用于局部加权匹配的性能优势。最后通过对比实验证明: 在不损失精度的前提下, 改进后的匹配代价整体计算时间提高了近 1/2。

关键词: 测地线距离; 加权窗口; 匹配代价; middlebury

Improved algorithm for local weighted matching cost computation

Yuan Man^{1),2)}, Liu Zhi¹⁾, Wang Fan¹⁾, Hu Tao²⁾

¹⁾(Institute of Surveying and Mapping, Information Engineering University, Zhengzhou 450052 China)

²⁾(78155 Troops, Chengdu 610036 China)

Abstract: Dense matching can be generalized as a process of energy minimization. The computation of matching cost is an independent step for Dense matching. We analyzed a recent well-performed weighted local matching algorithm, with which we proposed an improvement of core steps named the computation of geodesic distance. After introducing the advancements of the new algorithm in detail, we show that: nearly 1/2 time cost of matching cost computation was saved without loss of precision in our experiment.

Keywords: geodesic distance; weighted window; matching cost; middlebury

0 引言

在待匹配点周围定义一个邻域, 并利用邻域间的某种相关关系进行匹配代价的计算, 是匹配研究的重要方面。这一类局部方法在明德学院^[1]的划分中归属于匹配代价计算 (aggregate) 阶段, 该阶段所得到的结果通常被表示为 3 维空间上的匹配代价 $Cost(x, y, d)$ 。局部匹配代价的计算本身可以看做一种局部匹配, 也可以看做下步全局优化匹配的初始代价计算过程, 文献[2]对多种局部匹配代价计算方法进行精度和速度

上的比较, 认为选择一种良好的局部匹配代价计算方法将有助于提高全局优化算法最终的视差图生成质量和速度。

在这类局部匹配代价计算方法中, 共同的挑战就是如何选取特定点的“最优”局部邻域。这个“最优”的局部邻域应该有足够多的纹理细节以满足特征的独特性; 同时, 它又需要尽量减少其中可能具有不同视差的点。只有这样的窗口才能更好地概括特定点的独特性, 以提高匹配的鲁棒性。

因为固定大小和形状的窗口相关的种种不稳定性^[3]。Kanade 等人提出自适应窗口 (AW) 的方法^[4]。这种方法通过从一个较小的窗口开始向 4 个方向扩

收稿日期: 2010-05-28; 修回日期: 2010-09-05

第一作者简介: 袁满 (1982—), 男, 解放军信息工程大学测绘学院摄影测量与遥感专业硕士研究生, 主要研究方向为遥感影像的稠密匹配技术。E-mail: shinings@163.com。

张,过程中迭代计算每次扩张后整个窗口的不确定值(uncertainty),在达到最小的不确定值或一定的迭代次数后停止。这种方法相比固定窗口方法引入了较大的计算量,并且其中参数如何取定也很大程度上决定了最终匹配的质量。另外,这种方法最大的一个缺陷是:窗口的形状只能是矩形而不能任意改变。因此,并不能认为是真正意义上的自适应。如何实现窗口形状的自适应,成为其后很多学者研究的热点,如:文献[5-6]提出自己的窗口自适应方法,尽管在实现方法上不尽相同,但总的来说都具有较大的时间复杂度。文献[7-8]提出的自适应方法使用整数图像优化方法大幅提高了速度,但是算法效果过多依赖参数取定,使得通用性不佳,而且也没加入对窗口形状自适应的考虑。

Yoon 等人^[9]利用格式塔理论(gestalt principles)中关于人眼视觉的特点设计对窗口的加权模型:利用与中心像素点的颜色差值和距离计算矩形窗口内点的权值,即颜色差异越大,距离越大则权值越小。通过计算得到的权值掩模对图像的视差空间进行一次空间域卷积运算,即实现匹配代价的计算。这种利用距离和颜色差异加权的方式实现了一种窗口形状的自适应,且比起通常的窗口相关方法计算量并未显著增加。但是,在权值的距离计算中只是简单使用像素间的空间距离,这样使得距离项的使用仅仅体现了一种平滑截窗效果。对于视差断裂区域,这种方法效果并不好。

文献[10]在此研究的基础上,定义一种2维的颜色测地线距离(CGD),利用CGD取代空间距离计算得到的权值更好的表达同一视差范围内像素点的相近关系,有效克服使用空间距离的不足,从而取得了通过明德学院^[1]实验数据证明的“当前最好的

局部匹配算法”的效果,另外,利用CGD在图像分割^[11]和遥感影像DEM内插^[12]方面应用也取得了较好的效果。

因为 $G(V,E)$ 规模庞大,Dijkstra一类的最短路径算法对于CGD计算并不适用。文献[5]对CGD的计算利用的是一种行式近似的动态规划方法,整个过程需要多次迭代才能取得较好的近似结果。提出一种环式的动态规划方法计算CGD,使得计算过程仅正反两次遍历即可取得更优于行式近似的效果,而不再需要多次迭代,使得单个窗口中运算量减少2/3。

1 CGD 加权窗口匹配

1.1 CGD 加权窗口匹配步骤

计算局部窗口内任意像素点 p 到窗口中心像素点 c 的CGD可以被表示为

$$\text{CGD}(p,c) = \min d(\mathbf{P}) \quad (1)$$

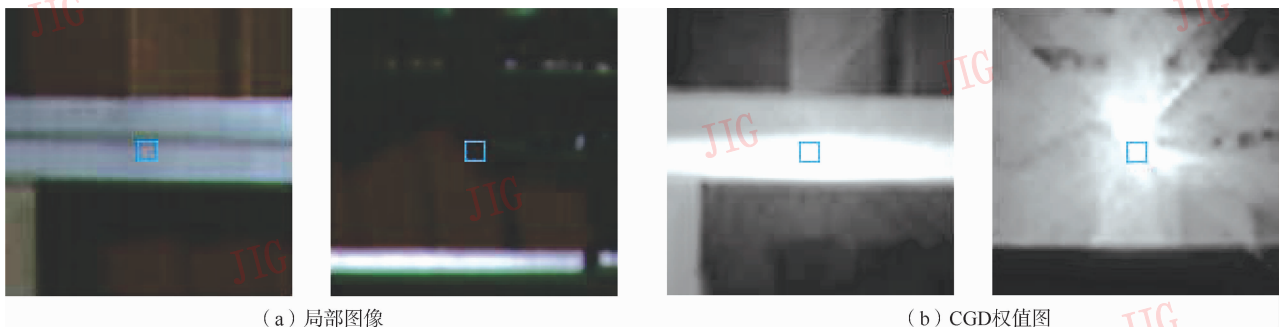
式中, \mathbf{P} 为 p 到 c 的所有可能路径的集合。而任意像素点间的路径被定义为一个8连通点的序列 $\{p_1, p_2, p_3, \dots, p_n\}$ 。路径的花销大小由相邻像素颜色差异之和决定,即

$$d(\mathbf{P}) = \sum_{i=2}^{i=n} d_c(p_i, p_{i-1}) \quad (2)$$

其中对于RGB图像,

$$d_c(p,q) = \sqrt{(R_p - R_q)^2 + (G_p - G_q)^2 + (B_p - B_q)^2}$$

直观来说,路径上相邻点颜色变化不剧烈则路径的花销相对较少。如果两点间存在至少一条花销较小的路径,则两点间即便在像平面上两点空间距离较大,它们的CGD也会相对较小。如图1所示:中心框表示中心像素点;亮度越大则CGD权值越



(a) 局部图像

(b) CGD权值图

图1 局部图像和对应CGD权值图

Fig. 1 Original local images and corresponding GCD weight map

大。可以看出,使用 CGD 来计算的窗口权值实现了窗口形状自适应的效果。

因此,由 CGD 决定窗口内单个像素 p 的权值 w ,以描述该像素点与中心像素点 c 的视差接近程度就更为合理。依文献 [4] 的做法, w 由式 (3) 计算得到:

$$w(p, c) = \exp\left(-\frac{\text{CGD}(p, c)}{K}\right) \quad (3)$$

式中 K 为根据窗口大小情况而自行选定的经验常数。

通过式 (4) 可以得到点 c 取视差 d 时的匹配代价:

$$\text{Cost}(c, d) = \sum_{p \in W} w(p, c) \cdot f(p, p - d) \quad (4)$$

式中 $f(p, q)$ 是 p 点和另一张影像上的对应视差为 d 的点 q 的颜色差值, W 是以点 c 为中心的窗口内像素点的集合。

在图像所有像素的 3 维匹配代价 $\text{Cost}(x, y, d)$ 计算完毕后,最后可以由简单的赢者通吃 (WTA) 挑选出每个像素点 $c(x, y)$ 对应最小匹配代价的 d 即为 c 点最终的视差值:

$$d_{\min} = \operatorname{argmin}_{d \in D} \text{Cost}(c, d) \quad (5)$$

式中, D 为 c 点全部可能视差值的集合。

1.2 行式 CGD 近似计算

文献 [5] 中,CGD 的计算是利用的一种逐行遍历的方法。如图 2 所示,以 c 为中心的 5×5 局部窗口。首先,将窗口内每个除中心点外的像素点 p 的路径花销 $C(p)$ 都初始化为一个较大的常数 T ,而中心点的花销初始化为零。然后从左上角 F 点开始,由左至右、由上至下逐点遍历计算路

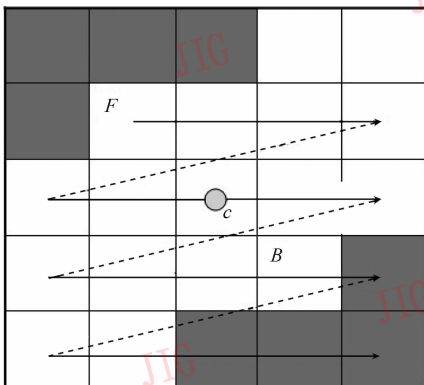


图 2 行式近似 CGD 计算

Fig. 2 Approximation of CGD in row major order

径花销,在达到最后一个点后从 B 点开始反方向进行,这样完成一次迭代。

对于窗口内任意一点 p 来说,其路径花销 $C(p)$ 是其允许决策集合 K_p 中的任意一点 q 的路径花销 $C(q)$ 与两点颜色差异之和的最小值,见下式

$$C(p) = \min_{q \in K_p} C(q) + d_c(p, q) \quad (6)$$

如图 2 所示,在正向进行时,除边缘点外, p 点的允许决策集合包括左、左上、上、右上 4 个方向的相邻点以及自身;反向进行时,是 p 点右、右下、下、左下 4 个方向相邻点以及自身。这样在经过多次迭代后,路径花销 $C(p)$ 即为 p 点的近似 CGD。

2 CGD 算法改进

行式 CGD 近似算法的遍历过程是逐行进行的,方向始终由左至右,允许决策集合的选取对于当前像素点的相对位置是固定的。一次正反向遍历后完成即完成一次迭代,这种迭代至少要进行 3 次以上才能取得较好的结果^[10]。

在行式 CGD 算法的基础上,用环式的遍历策略取代行式的遍历,并通过理论和实验证明这种遍历策略在不损失精度的前提下大幅提高了 CGD 搜索的效率。

2.1 环式 CGD 算法

环式 CGD 遍历开始前同样需要将窗口内每个除中心点外的像素点 p 的路径花销 $C(p)$ 都初始化为较大的常数 T ,中心点花销初始化为零。

如图 3 所示:从局部窗口的中心点的下方临近点 S 开始,由内向外的呈环状展开,最终遍历窗口内

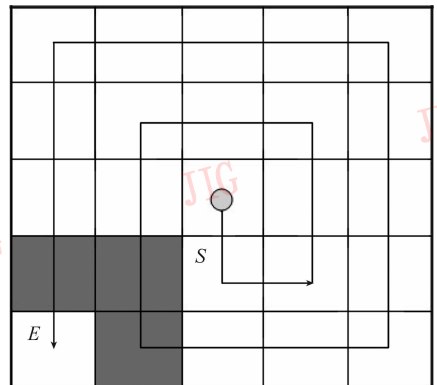


图 3 改进的 CGD 近似计算

Fig. 3 Approximation of the CGD in round major order

的所有像素点。每个当前点的允许决策集合 K_p 包括当前点本身以及其转移方向的后、左后、左、左前4个方向的临近点,在实际计算中这些点需要满足已经遍历过的条件。

具体对于起点 S 来说,允许决策集合除本身外只有窗口中心点;而终点 E 点临近的灰色点是它的允许决策集合。

在允许决策集合的选取方法确定以后,由内向外逐点向外展开遍历,并分阶段利用式(6)计算各点到中心点的路径花销,这是一个标准动态规划路径搜索的过程,不再赘述。

2.2 两种算法的分析和比较

从图3可以直观的看出,环式算法实现了一种

灵活的允许决策集合选取的机制:当处于中心点4个不同的方位时,当前点的允许决策集合总是偏向更接近中心点的位置。比起行式算法,环式算法能用更少的遍历步骤涵盖更多的可能路径。

如图4所示:(a)一(c)是行式算法不同遍历阶段的路径连通示意图;(d)一(f)是环式算法不同遍历阶段的路径连通示意图。灰色区域像素点表示没有通向中心点 c 的可能路径;以任意像素点 p 为中心的箭头表示: p 点与箭头所指方向相邻像素点 q 之间的路径花销在已经完成的遍历过程中被纳入到近似 CGD 的计算。换句话说,如果 p 点没有指向 q 点的箭头,则近似 CGD 的计算结果不可能经由 p 点到达 q 点,反之亦然。

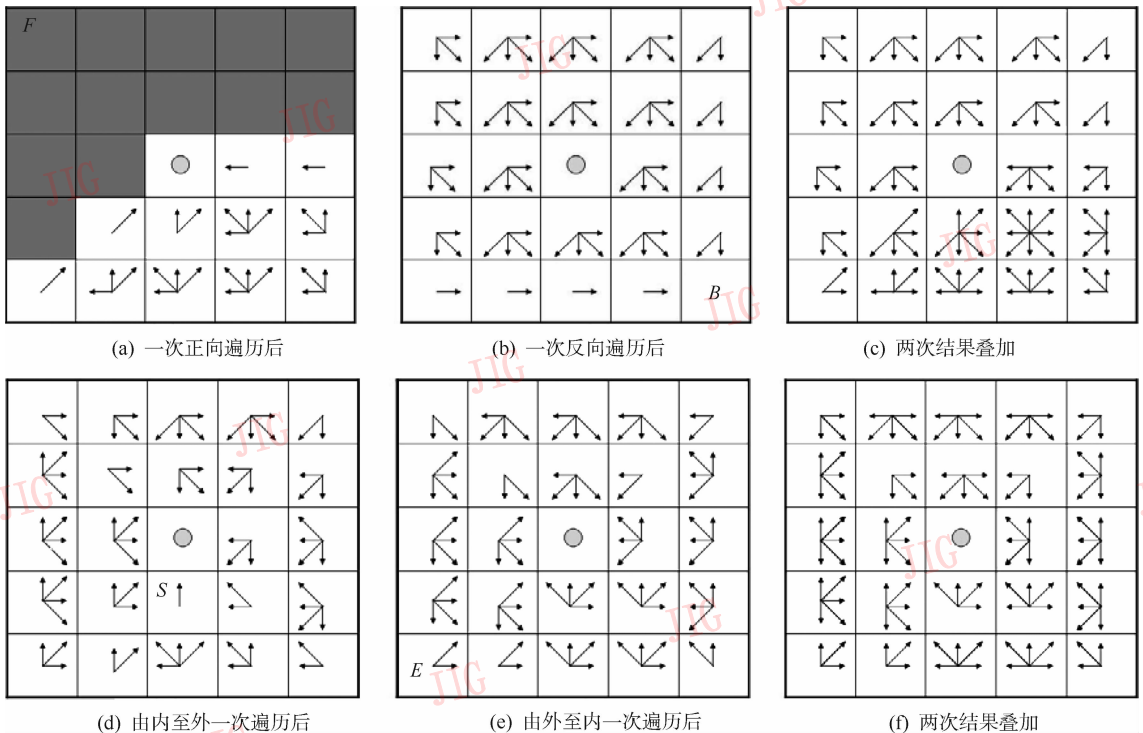


图4 行式和环式算法路径连通示意图

Fig. 4 Schematic diagrams for connected path of two algorithms

所以,CGD 的算法效率,在某种意义上可以被认为是由遍历的时间复杂度和在有限次遍历后窗口内任意相邻两点的可达到概率共同决定的。

环式算法的一次由内向外或由外向内的遍历和行式算法一次正向或反向遍历时间复杂度一样,均为 $O(|W|)$, W 是局部窗口像素点的个数。

通过图4的对比可以发现:如果只进行一次遍历,行式算法将无法计算窗口内全部像素点到中心点的近似 CGD,然而环式算法却可以很好地

实现。如果进行正反向各一次遍历,行式算法会在某些部分出现一种整体路径“断裂”的现象,如图4(c)中的中心像素点所在行与上一行之间出现整体的“断裂”,这种整体“断裂”将使近似 CGD 与真值之间的误差大幅增加,也是因为这个原因,使得行式遍历需要多次迭代进行。相比之下,环式算法在两次遍历后仅在拐角点处损失部分可能路径,这种路径损失比并不会出现整体路径“断裂”。

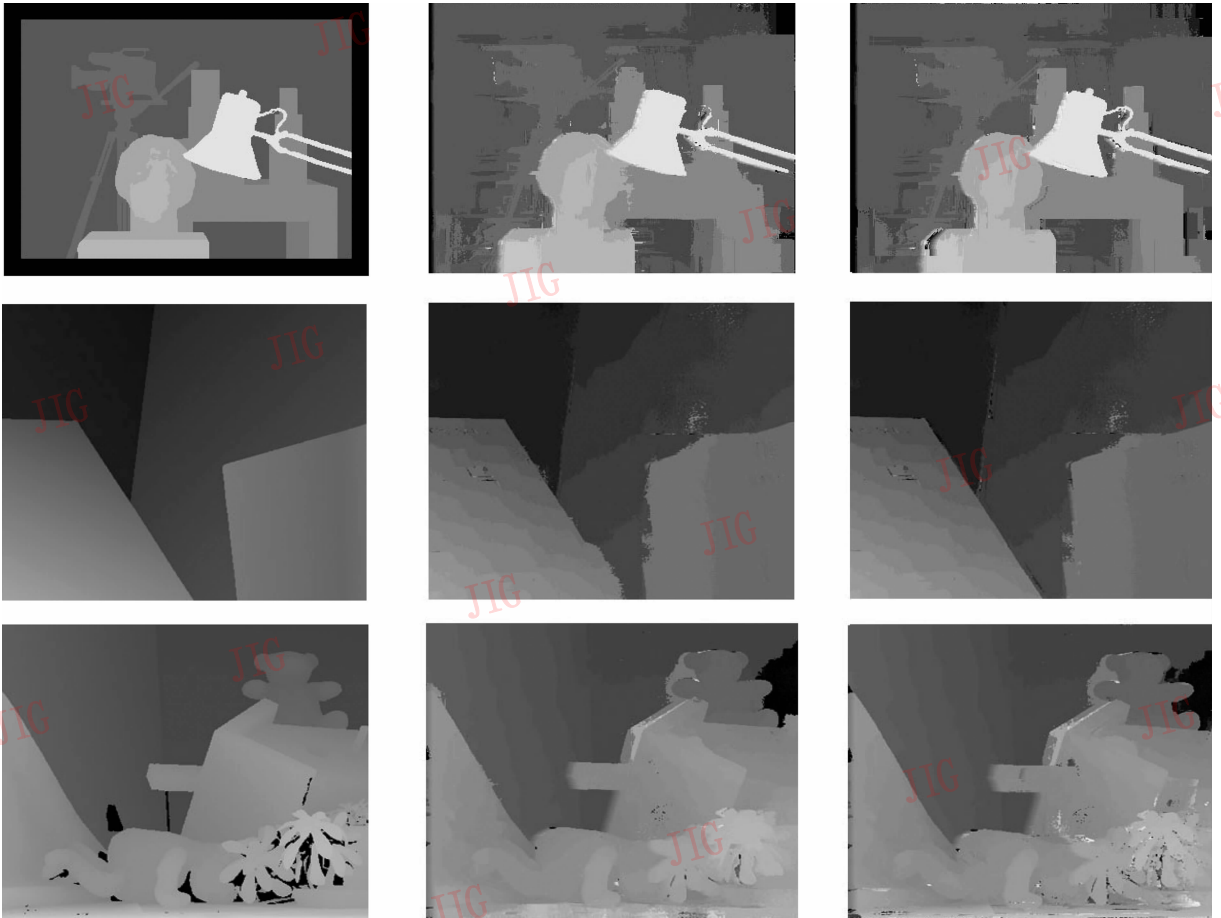
3 实验结果与分析

实验选用 middlebury 的 3 组代表性数据 tsukuba、venus 和 teddy 分别使用提出的环式 CGD 近似算法和文献[5]的行式 CGD 近似算法进行匹配代价的计算,在此基础上利用 WTA 求得视差图,作为最终数据进行对比。

具体实验参数为:视差图的视差层次 tsukuba、venus 和 teddy 分别为 16、8 和 4。窗口大小都设为 31, K 设为常数 50。另外,行式算法参照文献[5],每个窗口的 CGD 计算进行了 3 次迭代即正反 6 次遍历。环式算法进行了 1 次迭代即由内至外和由外至内各一次遍历。运行环境为 P4(3.0 GHz)。均未使用并行优化。

因为两种方法均属匹配代价计算阶段,为了直接的对比两种方法在计算匹配代价上的性能优劣,没有采用文献[5]中对比左右视差图的遮挡去除方法对视差图的计算结果进行再优化。

最后,通过实验的得到的左像视差图,如图 5 所示。两种方法的计算时间和视差错误率对比在表 1 列出。利用环式算法计算 CGD 进行加权窗口匹配代价的计算在略有精度提高的前提下,匹配代价的计算时间大大减少。匹配代价的计算占用机时主要在于权值计算和加权相关两个阶段,环式算法将权值计算需要的迭代次数减少了 2/3,而整个匹配代价的计算实际减少了近 1/2 的运算时间。有理由相信在利用全局优化的方法后视差图效果将进一步提升。



(a) 真实视差图

(b) 行式CGD近似算法匹配结果

(c) 环式CGD近似算法匹配结果

图 5 行式和环式算法匹配结果

Fig. 5 Matching results of two algorithms

表1 两种算法对比

Tab.1 Compare of two algorithms

算法	tsukuba		venus		teddy	
	运行时间/s	视差错误率/%	运行时间/s	视差错误率/%	运行时间/s	视差错误率/%
行式算法	354.39	5.35	576.30	6.18	978.09	9.92
环式算法	178.16	4.11	294.17	4.62	604.77	8.70

4 结论

利用局部窗口 CGD 进行加权匹配代价计算,可以很好地实现窗口大小和形状的自适应,提高匹配代价的计算精度。文献[5]证明这类方法的有效性。相较起行式 CGD 计算方法,提出一种环式的 CGD 计算方法,在不损失精度的前提下,使得局部窗口 CGD 的计算不再需要多次迭代进行,减少了近 2/3 的运算时间。并通过实验验证了算法的优势。

今后的研究主要将集中于以下两个方面:

1) 相较普通窗口匹配代价计算,加权窗口匹配代价计算因为不能进行最大程度上减少计算冗余度的滑动窗口优化^[10-13],具有实现速度上的劣势。如何优化算法速度将是今后算法改进的重点;

2) 窗口的自适应该是形状和大小的自适应。环式搜索因为其由内至外的特点,在实现窗口大小自适应上有独到优势。实现在纹理丰富的区域使用较小的窗口,在纹理缺乏的区域使用较大的窗口,并实验应用于一些最新的稀疏匹配方法^[14],以期取得较好效果。

参考文献 (References)

- [1] 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. <http://www.middlebury.edu/stereo/>.
- [2] Wang L, Gong M W, Gong M L, et al. How far can we go with local optimization in real-time stereo matching [C] // Proceeding of the 3th International Symposium on 3D Data Processing. Chapel Hill, USA: IEEE Computer Society, 2006: 129-136.
- [3] Delon J. Fine Comparison of Images and Other Problems [D]. Paris ; Paris-Dauphine University, 2004.
- [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] Bobick A, Intille S. Large occlusion stereo [J]. International Journal of Computer Vision, 1999, 33 (3): 181-200.
- [6] Fusiello A, Roberto V, Trucco E. Efficient stereo with multiple windowing [C] // Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition. San Juan, Puerto Rico: IEEE Computer Society, 1997: 858-863.
- [7] Boykov Y, Veksler O, Zabih R. A variable window approach to early vision [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20 (12): 1283-1294.
- [8] Veksler O. Fast variable window for stereo correspondence using integral images [C] // Proceedings of the 2003 Conference on Computer Vision and Pattern Recognition, America. Princeton, NJ, USA: IEEE Computer Society, 2003: 556-561.
- [9] 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.
- [10] Hosni A, Bleyer M, Gelautz M, et al. Local stereo matching using geodesic support weights [C] // Proceedings of the 16th International Conference on Image Processing. Cairo, Egypt: IEEE Computer Society, 2009: 2093-2096.
- [11] Hosni A, Bleyer M, Gelautz M. Image segmentation via iterative geodesic averaging [C] // Proceedings of the 5th International Conference on Image and Graphics. Washington, DC, USA: IEEE Computer Society, 2009: 250-255.
- [12] Facciolo G. Variational Adhesion Correction with Image Based Regularization for Digital Elevation Models [D]. Uruguay: Universidad de la República, 2005.
- [13] Faugeras O. Real Time Correlation Based Stereo: Algorithm Implementations and Applications [R]. Paris, INRIA: Technical Report, 1996.
- [14] Wang H, Yan S C, Liu J Z, et al. Correspondence propagation with weak priors [J]. IEEE Transactions on Image Processing, 2009, 18 (1): 140-150.