

一种自适应多窗口的立体匹配算法

曾凡志 鲍苏苏

(华南师范大学计算机科学系 广州 510631)

摘要 提出了一种新型的自适应匹配窗口的方法来解决低纹理图像在立体匹配中容易出现误配的现象。该算法采用 8 个相同的窗口根据图像的平滑情况往 8 个方向选择适当的支撑区域,且其非常适合并行处理。然后根据上述步骤得到的结果提出一种自适应的区域生长法对误配和遮挡区域进行后处理。实验结果表明,该方法无论在低纹理区域还是高纹理区域都有较好的实验结果,并且运行时间与传统的固定窗口局部方法相差不大,适合实时处理。

关键词 立体匹配,低纹理,自适应多窗口,自适应区域生长,并行处理,实时匹配

中图分类号 TP391 **文献标识码** A

Adaptive Multiple Windows Stereo Matching Algorithm

ZENG Fan-zhi BAO Su-su

(Department of Computer Science, South China Normal University, Guangzhou 510631, China)

Abstract New kind of adaptive multiple windows stereo matching algorithm was presented, which can solve that easily appeared with error matching in textureless image in stereo matching. The algorithm adopts 8 same windows according the smooth situation of images to find appropriate support area by 8 direction, and this method is very suitable for parallel processing. Then presented a kind of adaptive region growing algorithm to solve that the obtained result by the above steps, which leaves something to be desired. Experimental results show that this method is whether in textureless regions or texture regions has greatly improved, and running time close to the traditional local fixed window method, which is suitable for real-time processing.

Keywords Stereo matching, Textureless, Adaptive multiple windows, Adaptive region growing, Parallel processing, Real-time matching

立体匹配是从两幅或者多幅视图中找到相应的匹配点,然后利用几何的方法求解三维空间的坐标位置,以达到三维重建的效果。立体匹配是视觉三维重建的关键步骤,也一直是立体视觉的热点。由于存在噪声、遮挡、透明、低纹理、重复纹理以及透视失真等影响,使得立体匹配非常困难。传统的立体匹配方法^[1]主要分为全局方法^[2,3]、半全局方法^[4]和局部方法^[7-11],但大多都是运行时间和效果不能兼得。其中,全局方法效果好,但时间复杂度最大;局部方法时间复杂度小,但效果一般。由于全局方法都建立在局部方法的数据匹配项上,因此研究局部方法显得更加基础与重要。

本文主要研究实时算法,所以着重对局部方法进行改进。传统的局部方法在高纹理区域有良好的匹配效果,但在平滑区域、遮挡区域以及物体边界深度不连续区域效果都不甚理想。在匹配过程中,解决平滑和遮挡区域问题需要大窗口,以包含更多的信息,而在物体边界深度不连续区域则需要小窗口避免前景增肥。于是很多学者提出了自适应窗口,其虽然效果改良了一些,但时间复杂度却损失很大。传统的局部方法(Fixed window)非常适合并行计算及 Box-Filtering^[5]和 Integral Images^[6](积分图)加速,而改进后的部分自适应窗口算法^[7-11]只能串行计算,并不能利用 Box-Filtering 和 Integral

Images 算法加速,或者利用了很多额外的计算。本文为了解决此类问题,提出了一种与传统局部方法时间复杂度相一致的自适应窗口匹配算法。经过以上步骤得到的视差图往往不是最好的,在后处理中本文提出了一种适合该视差图进行修正的快速自适应区域生长法算法对其来进行修复。

本文假设我们的图像数据都是经过校正后对极线在同一行,第 1 节先介绍本文的预处理方法;第 2 节介绍提出的一个新型的自适应窗口匹配算法;第 3 节提出一种适合该视差图进行修正的快速自适应区域生长法算法;第 4 节给出实验结果;最后给出实验结论。

1 前后处理和视差代价计算定义

1.1 预处理

为了方便处理,该算法所处理的数据是灰度图。为了避免光照等噪声的影响,对图像做了图像亮度归一化并加强了纹理的预处理,对于每个像素 I_c 都用以下公式进行计算:

$$I_c = \min(\max(I_c - \bar{I}_c, -I_{ap}), I_{ap}) \quad (1)$$

式中, $\bar{I}_c = \frac{\sum_{i=-n}^{i=n} \sum_{j=-n}^{j=n} I(x+i, y+j)}{(2 \times n + 1)^2}$, x, y 为 I_c 的坐标, n 为窗口的半径, I_{ap} 是大于 0 的常数。

本文受广东省部产学研结合项目(2009B080701077)资助。

曾凡志(1985-),男,硕士,主要研究方向为计算机视觉等, E-mail: zengfanzhi512@qq.com; 鲍苏苏 男,教授,硕士生导师,主要方向为模式识别、医学图像等。

1.2 后处理

对于第2节算法求解视差图后,为了得到更加可靠的视差图,利用探斑技术对视差图进行滤波。主要原理是:

$$d(i,j) = \begin{cases} 0, \max(|d(x,y) - d(i,j)|) > T \\ d(i,j) \end{cases} \quad (2)$$

式中, $d(i,j)$ 为视差图。 T 为某个阈值,一般取4。 $(x,y) \in W(i,j)$, $W(i,j)$ 为以 (i,j) 为中心的一个窗口,边长一般为9。

1.3 窗口匹配代价函数

对于梯度图,我们用sobel算子求解,并采用均值滤波,用 $g(x,y)$ 表示求解得到的梯度图。所采取的匹配代价函数是时间复杂度比较小的绝对误差累加(SAD),它的定义为:

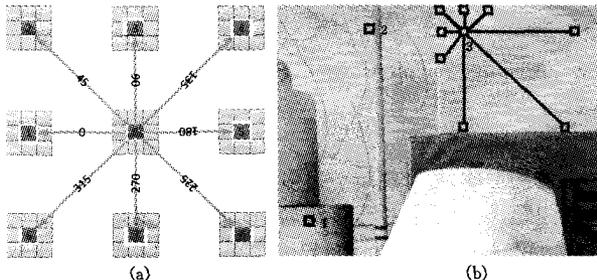
$$C = \sum_{i=-n}^n \sum_{j=-n}^n |I_1(x+i, y+j) - I_2(x+i, y+j)| \quad (3)$$

式中, I_1 和 I_2 是左图像与右图像的灰度值。 n 为以 (i,j) 为中心的窗口半径。

2 新型的自适应窗口匹配算法

2.1 自适应窗口的描述

如图1所示,本算法定义了9个都是 $size \times size$ 的支撑窗口,其中中间那个和传统方法一样,该窗口以需要求解视差的中心点 I_0 为中心;其余8个是以 I_0 为出发点,根据图像的平滑程度(即根据梯度图)往8个方向(上、下、左、右、左上、左下、右上、右下,参考图1(a))搜索,直到遇到梯度比较大的地方再回退一点(因为梯度比较大的地方很有可能是深度不连续的地方),再将这8个点($I_k, k=1, 2, \dots, 8$)为中心点的窗口作为 I_0 附属支撑窗口。



(a)代表支撑窗口,总共有9个。(b)代表不同像素点的不同支撑窗口,标号3那个点是平滑区域支撑区域,9个支撑窗口伸展得非常,标号1和标号2分别代表边界和高纹理区域,它们的9个窗口都交叠在一起

图1

本文采用去掉统计数据的最大与最小代价:

$$C_{SAD} = \sum_{k=0}^8 C_{I_k} - \max(C_{I_k}) - \min(C_{I_k}) \quad (4)$$

2.2 支撑窗口坐标的计算

先描述8个方向的支撑窗口的坐标记录算法,此算法主要运用了动态规划算法进行加速。为了简洁,我们只用公式进行上、下描述, $g(x,y)$ 表示梯度图,其他6个方向以此类推,表达式如下:

$$T(down, i, j) = \begin{cases} j, & g(i, j) > th \\ T(down, i, j-1), & g(i, j-1) \leq th \& g(i, j) < th \\ k, k = \min\{y, g(i, y) > th \& j \leq y \leq h\} \end{cases} \quad (5)$$

$$T(up, i, j) = \begin{cases} j, & g(i, j) > th \\ T(up, i, j-1), & g(i, j) \leq th \end{cases} \quad (6)$$

式中, $T(x, i, j)$ 为 (i, j) 8个方向窗口位置记录表, x 为8个方向。式(5)中的 h 为图像的高度, k 代表从 $y=j$ 值向下搜索

直到满足 $g(i, y) > th$ 或 $y=h$ 时的下标。

在 $T(x, i, j)$ 求解后我们需要对其调整,因为8个方向在伸展中,往往都是伸展到深度不连续处,包含了很多不合理信息,所以我们让8个方向回退半个窗口,大小用 n 表示。这里只对一个向上方向 up 进行描述,其他7个依次递推,公式如下:

$$T(down, i, j) = \begin{cases} T(down, i, j) - n, & T(down, i, j) - n > j \\ j, & T(down, i, j) + n \leq j \end{cases} \quad (7)$$

利用聚集算法分析技巧可以得到算法的时间和空间复杂度都为 $O(h \times w)$ 。

2.3 支撑窗口代价计算

由于在多个窗口中存在很多窗口在不同视差 d 的匹配代价被多次使用,因此为了加快算法的深度,我们采用空间换时间策略,用表 $C(d, i, j)$ 来记录每一点 (i, j) 在每一个视差 d 的匹配代价(根据式(3)计算),以便在求解视差时,直接查表就可以得到相应视差的代价。在该计算中利用并行技术和Box-Filtering技术来加速,填表的时间空间复杂度都为 $O(h \times w \times d)$ 。

2.4 视差的求解算法

有了上面的两个表 $T(x, i, j)$ 和 $C(d, i, j)$,我们就可以采用胜者为王策略(WTA)算法利用式(4)来求解视差了。对于图片的每一点 (i, j) ,都按照图2所示步骤实现。

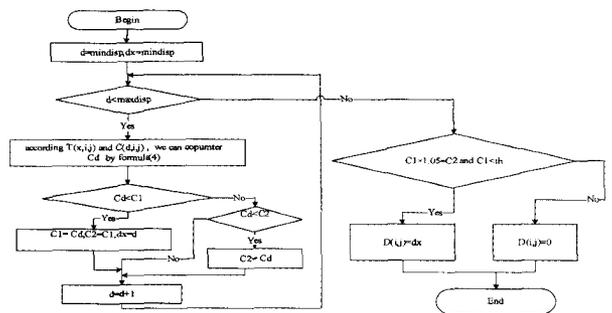


图2 视差求解的流程图

2.5 算法的优点分析

该算法除了空间复杂度为 $O(h \times w \times d)$ 比传统固定窗口大以外,其他方面复杂度都是一样的,并且其非常适合并行处理和实时计算。此算法还有以下优点:

1)传统的自适应窗口算法大多是针对某个数学模型根据平滑情况来调整支撑窗口大小或形状,时间复杂度非常大。如果选择最优支撑窗口则为NP问题,即使简化后也很难用并行和Box-Filtering加速,时间复杂度大多为 $O(h \times w \times d \times (n+f))$,其中 n 为支撑窗口的像素个数, f 为寻找最优支撑窗口的计算次数。

2)在纹理区域,窗口延伸非常小,甚至9个都是以 I_0 为中心的窗口,这样就和传统方法一样可以在高纹理或深度不连续处保证匹配点是高度置信的,并且能避免前景物体变肥。

3)本算法对于那些大面积的高度平滑物体优势明显。对于大面积的平滑物体,最好就是分割得到它的边界特征值,再进行匹配比较,而本算法恰好就是利用了它的边界特征值。由式(7)可知本算法8个附属窗口不会伸展到边界的中间,不会同时包含不同深度信息,而传统自适应窗口匹配算法有时候会引进一些其他不同深度的信息以致误差。

3 自适应区域生长修正算法

在进行双目匹配的时候,往往都依赖人类强大的分割能

力,来提取那些重要的特征点(如角点、颜色、边缘以及物体之间的相互关系),然后再匹配。但分割是一个研究难点,有些效果好的,时间复杂度太大了,比如 *meanshift*^[12] 算法。在后处理中提出一种适合本文研究内容的一个快速自适应区域生长的分割算法来对那些误配或没有匹配的点进行修正。

3.1 算法的描述

根据第2节所描述的算法得到除了那些视差不可信点(我们定义视差为0),其他都为视差可信点。所以我们将所有的不可信点作为种子点进行区域生长,并统计该生长区域的可信点的视差分布,然后根据统计信息对分割区域内像素视差进行修正。算法描述如下:

Step1 将初始种子点压进队列。

Step2 如果队列已空或队列的大小大于 *maxNum*,转到 Step4。

Step3 将队列头的像素点弹出作为当前点,并遍历周围相邻的8个像素点,如果没有被遍历过并且和当前点的值差小于阈值 *th2*,就将其入队并且如果是可信点视差,就统计其视差直方图。

Step4 如果区域生长的像素个数小于 *minNum* 或区域生长的像素中的可信像素小于阈值 *th1*,并且 *th2* 小于阈值 *th3*,*th2* 加1,在 Step4 中如果我们统计的可信点大于某个百分比阈值 *th3*,*maxNum* 减去一个常数 *c*,返回 Step1。否则结束算法。

Step3 统计分割区域可信点视差和灰度值的分布,它的结果直接影响修复的效果。对可信点视差的直方图 [*mindisp*,*maxdisp*] 进行统计,将区间数值最大的视为该区域最可信的视差 *creD*。然后,根据式(8)更新该区域生长的所有像素视差。

$$d(i,j) = \begin{cases} creD, & D(i,j)=0 \text{ 或 } |d(i,j)-creD| > T \\ creD+f(d(i,j)-creD), & \text{其它} \end{cases} \quad (8)$$

式中,*I(i,j)*表示(*i,j*)的灰度值,*f(x)*为一个非递减函数。

Step4 中符合结束算法判断条件,只有两种情况:

1) 分割区域足够大,大于 *minNum* 个像素,并且有足够的可信点。因为只有这样才能给我们足够的信息,对分割区域进行修复。

2) 当分割阈值 *th2* 大于某个阈值 *th3* 时,直接进行更新,因为如果再把阈值 *th2* 调大会导致分割出错。

如果 Step4 不符合结束条件,把分割阈值 *th2* 调大重新运行区域生长算法,因为这样可以分割更大的区域。在 Step4 中如果我们统计的可信点大于某个百分比,则可以推断刚才分割的区域为高纹理区域,因为如果是高纹理区域生长点会很大,它很有可能不符合 Step4 结束条件,所以判断刚才分割的区域为高纹理区域。于是将队列的最大数目 *maxNum* 减去一个常数 *c*,这样一方面可以加快算法速度,另一方面避免分割区域太大,而造成分割失败。由于已经把分割阈值 *th2* 调大,因此一旦分割到大片平滑区域就很容易把高纹理区域归纳为平滑区域而造成算法失败。

3.2 本算法的优点

效果:本算法结合图像数据本身的特殊性和一般性,利用多个自适应的阈值调节,从而得到了比较好的实验结果。

算法时间复杂度:由于每个像素在生长算法中最多只会被纳入分割区域 $2 \times th2 + 1 \leq 2 \times th3 + 1$ 次操作,而 *th3* 为一常数,因此算法的时间复杂度为 $O(h \times w)$ 。设想如果某像素

pixel 的值为 *val*,和 *pixel* 联通所有灰度值为 $val1 \in [val - th2, val + th2]$ 并且视差值为0(即不可信视差点)的像素集合 *U* 为生长种子点的时候,最多只会对 *pixel* 操作一次。

证明:假设像素 $x \in U$ 和像素 $y \in U$,当 x, y 分别为种子点时,如果都没生长到 *pixel*,则对 *pixel* 有0次操作;如果都可以生长到 *pixel*, x 和 y 就是联通的,所以只会对 *pixel* 进行一次操作。所以对所有点 $z \in U$ 为种子点时,如果能生长到 *pixel* 点且为集合 *P* 时,*P* 是联通的,只会对 *pixel* 操作一次;如果生长不到 *pixel*,它们对 *pixel* 的操作为0次。所以 *pixel* 最多只会被操作 $2 \times th3 + 1$ 次。

4 实验结果与分析

为了客观评价本算法的性能,主要评价方法依据文献[4]标准,从网址 <http://www.middlebury.edu/stereo> 获得标准数据进行评测。为了评价本算法的通用性,算法对高纹理图像、低纹理以及遮挡图像进行测评。算法所用的都是灰度,没有利用到彩色信息。

4.1 实验效果

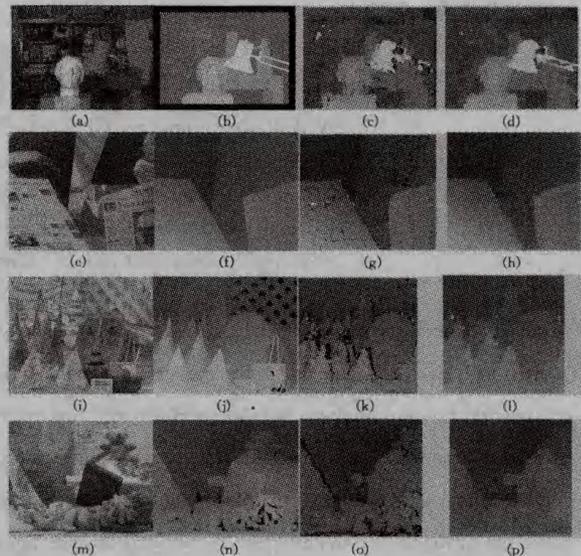


图4 由上到下分别为 Tsukuba、Venus、Cones 和 Teddy 测试图

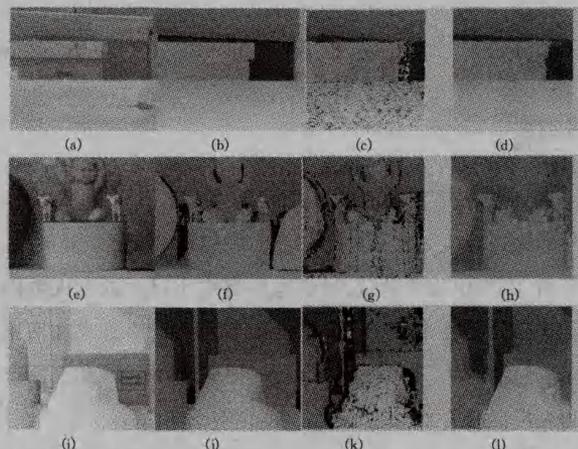


图5 由上到下分别为 wood、baby 和 Lampshade 测试图

实验结果如图4、图5所示,从左往右分别是原图、相应标准视差图、自适应窗口匹配算法求解图、自适应窗口匹配算

(下转第558页)

- [7] Soille P. Morphological Image Analysis: Principles and Applications(Second edition)[M]. Springer-Verlag, 2008
- [8] Salembier P. Morphological Multi-scale Segmentation for Image Coding[J]. Signal Processing, 1994, 38:359-386
- [9] Yaakov T, Amir A. Automatic Segmentation of Moving Objects in Video Sequences; A Region Labeling Approach [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2002, 12(7);597-612
- [10] Haris K, Efstratiadis S N, Maglaveras N, et al. Hybrid Image Segmentation Using Watersheds and Fast Region Merging[J]. Image Processing, 1998, 7(12); 1684-1699
- [11] Malpica N, Ortuno J E, Santos A. A multichannel watershed-based algorithm for supervised texture segmentation[J]. Pattern Recognition Letters, 2003, 24; 1545-1554
- [12] 郭皓. 中国近海赤潮生物图谱[M]. 北京:海洋出版社, 2004
- [13] 马东,程敬之,李斌. 细胞图像的彩色纹理分析方法研究[J]. 西安交通大学学报, 2000, 34(1); 108-110
- [14] Ridler T W, Calvard S. Picture threshold Picture thresholding using an iterative selection method[J]. IEEE Transactions on Systems, Man and Cybern, 1978, SMC-8; 630-632

(上接第 521 页)

法+自适应区域生长修正算法求解。

4.2 性能评价

文献[1]中定义了 2 个重要统计指标并对各种算法进行定量比较分析,算法也是根据这几个指标对 Tsukuba、Venus、Cones 和 Teddy 最终视差图和标准视差图进行测评比较,而其他图只看效果。这 2 个指标分别是 B_0 (非遮挡区域错误百分比)、 B_D (视差不连续区域错误匹配百分比)。

由于本文的算法非常适合并行,因此算法效率依赖硬件(多核 CPU 或 GPU),并且文中已对算法时间复杂度进行了分析,所以不再对算法的时间效率进行详细分析,只对本算法和其他算法进行一个不太系统的比较。本程序是用 C++ 基于部分 openCV 库编写的,并在 CPU 配置为 intel® Core Duo CPU P8700、主频为 2.53 的情况下运行得到实验结果。下面的其他算法的性能结果来自于 <http://www.vision.deis.unibo.it/spe/SPEresults.aspx>。

表 1 定量分析对比表

Algorithm	Tsukuba nonocc	Tsukuba disc	Venus nonocc	Venus disc	Cones nonocc	Cones disc	Teddy nonocc	Teddy disc	Time Teddy mm:ss
Our result	6.25	13.13	4.13	25.75	10.01	21.59	10.11	24.71	<2s
Locally Consistent+ Fixed Window ^[4]	3.07	9.63	0.66	5.11	10.60	21.80	5.30	11.60	00.15
Adaptive weight ^[5]	4.66	8.65	4.61	13.30	12.70	22.40	5.50	11.90	20.35
Variable windows ^[6]	3.12	12.40	2.42	13.30	17.70	25.50	21.20	27.30	00.26
Multiple windows(9W) ^[7]	7.60	25.70	6.18	29.00	18.00	35.60	11.80	27.10	00.14
Shiftable windows ^[1]	9.58	14.40	9.66	16.50	23.60	31.20	24.40	33.60	00.05
Segmentation based ^[11]	2.25	8.87	1.37	9.40	12.70	24.80	11.10	20.10	05.14
Fixed window	9.58	27.10	10.60	42.50	25.10	42.40	19.70	36.00	<1s

结束语 实验结果表明,本文的算法效果不比传统的变窗口或多窗口的算法差,且时间复杂度比相关算法大有改进,非常适合并行处理。本算法虽然无法和半全局或全局方法以及某些局部算法效果相比,但是在实际应用中我们常常需要实时处理,这是本算法的优势;同时本算法很容易和其他半全局或全局方法结合在一起,并且不会影响时间复杂度。相比复杂背景的场景,本算法更加适合处理低纹理平滑等场景。

参考文献

- [1] Scharstein D, Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms[J]. Int. Jour. Computer Vision, 2002, 47(1/2/3); 7-42
- [2] Kolmogorov V, Zabih R. Computing visual correspondence with occlusions using graph cuts[C]//ICCV. 2001
- [3] Sun Jian, Zheng Nan-ning, Shum H Y. Stereo Matching Using Belief Propagation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(7); 787-800
- [4] Kim J, Lee K, Choi B, et al. A dense stereo matching using two-pass dynamic programming with generalized ground control points[C]//Proc. Conf. on Computer Vision and Pattern Recognition(CVPR 2005). 2005; 1075-1082
- [5] Mc Donnell M. Box-filtering techniques [J]. Computer Graphics and Image Processing, 1981, 17; 65-70
- [6] Crow F. Summed-area tables for texture mapping [J]. Computer Graphics, 1984, 18(3); 207-212
- [7] Mattocchia S. A locally global approach to stereo correspondence [C]// IEEE Workshop on 3D Digital Imaging and Modeling (3DIM 2009), 2009
- [8] Gerrits M, Bekaert P. Local Stereo Matching with Segmentation-based Outlier Rejection [C]//Proc. Canadian Conf. on Computer and Robot Vision(CRV 2006). 2006; 66-66
- [9] Veksler O. Fast variable window for stereo correspondence using integral images [C]//Proc. Conf. on Computer Vision and Pattern Recognition(CVPR 2003). 2003; 556-561
- [10] Hirschmuller H, Innocent P, Garibaldi J. Real-time correlation-based stereo vision with reduced border errors [J]. Int. Journ. of Computer Vision, 2002, 47; 1-3
- [11] Gerrits M, Bekaert P. Local Stereo Matching with Segmentation-based Outlier Rejection[C]//Proc. Canadian Conf. on Computer and Robot Vision(CRV 2006). 2006; 66-66
- [12] Comaniciu D, Meer P. Mean shift; a robust approach toward feature space analysis [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(5); 603-619