基于限制搜索空间的快速立体匹配方法

门朝光,边继龙,李 香

(哈尔滨工程大学 计算机科学与技术学院,哈尔滨 150001)

摘 要:为快速而准确地得到稠密视差图,提出了一种基于限制搜索空间的动态规划立体匹配算法。该算法以动态规划立体匹配方法为基础,通过初始匹配序列限制搜索空间以减少搜索变量个数。同时,提出一种基于自适应权重的多窗口累积策略来提高匹配精度,并在平滑性限制中引入亮度梯度以避免在物体边界的视差不连续处产生过度惩罚。实验结果表明,该匹配算法在匹配速度和匹配精度上都有很大的提高,是一种简单有效的立体匹配算法。

关键词:计算机应用;立体匹配;动态规划;视差估计;搜索空间

中图分类号:TP391.41 **文献标志码**:A **文章编号**:1671-5497(2012)02-0423-06

Fast stereo matching algorithm based on limited search space

MEN Chao-guang, BIAN Ji-long, LI Xiang

(*College of Computer Science and Technology, Harbin Engineering University, Harbin* 150001, *China*)

Abstract: A fast stereo matching algorithm based on dynamic programming on limited search space is proposed to attain an accurate dense disparity map. The proposed algorithm is built on the base of dynamic programming for stereo matching and it limits the search space by an initial match sequence to reduce the number of vertices to be expanded. The multi-window aggregation strategy based on adaptive weight is proposed to improve the matching accuracy. The brightness gradient is integrated into the smoothness constraints to avoid over-punishment on disparity discontinuity on object boundary. Experimental results show that the matching speed and accuracy of the proposed algorithm are greatly improved, and the algorithm is simple and effective for stereo matching.

Key words: computer application; stereo matching; dynamic programming; disparity estimation; search space

目前根据文献[1]提出的分类标准,立体匹配算法可大致分为局部立体匹配算法[2-3]和全局立体匹配算法[4-10]两类。局部立体匹配算法的优点是时间复杂度低,易于实现,但缺点是视差准确率较低;全局立体匹配算法的优点是视差准确率较高,但时间复杂度也较高。基于动态规划的立体匹配算法属于全局匹配算法,它把对应点搜索问题转化为在视差空间中根据次序性约束搜索最优成本路径问题,而且可以在匹配的同时检测遮挡,最终生成准确率较高的视差图。

为减少动态规划立体匹配算法的搜索空间,同时提高算法的匹配精度,本文提出了一种基于

收稿日期:2010-10-28.

基金项目:国家自然科学基金项目(61100004);中央高校基本科研业务费专项资金项目(HEUCF100607).

作者简介:门朝光(1963-),男,教授,博士生导师.研究方向:可信计算,图像处理.
E-mail:menchaoguang@hrbeu.edu.cn

通信作者:边继龙(1982-),男,博士研究生.研究方向:立体匹配,三维重建.E-mail:bianjilong@hrbeu.edu.cn

限制搜索空间的动态规划立体匹配算法,该算法通过初始匹配序列限制搜索空间规模。同时,提出了一种基于自适应权重的多窗口累积策略计算匹配成本以提高匹配精度,并在平滑性限制中引入亮度梯度以避免在物体边界处产生过度惩罚。

1 建立视差空间

1.1 视差空间图

视差空间图[7](Disparity-space image,DSI)为三维数据结构,在基于动态规划的立体匹配中经常使用此数据结构研究匹配中的遮挡和对应问题。DSI 是视差空间的一种表示方法,反映视差范围内可能对应点之间的匹配代价,其中第 y 行 DSI 的数学表达式为

$$DSI_y^L(x,d)=\| I_L(x,y)-I_R(x-d,y)\| \tag{1}$$

式中:d 为视差;$I_L(x,y)$、$I_R(x-d,y)$ 分别表示左右图像。

为减少噪声对匹配的影响,一般选择一个支撑窗口累积匹配代价,其数学表达式为

$$DSI_y^L(x,d)=\sum_{s=-c_x}^{w_x-c_x}\sum_{t=-c_y}^{w_y-c_y}\| I_L(x+s,y+t)-I_R(x-d+s,y+t)\| \tag{2}$$

式中:$w_x\times w_y$ 表示窗口大小。

1.2 累积策略

虽然固定窗口策略可以减少噪声对匹配的影响,但是这可能致使支撑窗口跨越物体边界而导致算法在深度不连续处产生错误匹配。为解决此问题,本文提出一种基于自适应权重的多窗口策略来累积匹配代价,在每个参考像素上使用 9 个不同形状的窗口和相应的自适应权重函数累积匹配代价,这些窗口位置和形状如图 1 所示。

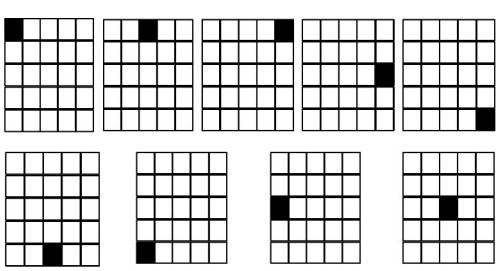


图 1 多窗口策略

Fig. 1 Multiple window strategy

多窗口是指通过改变窗口中心位置而形成的 9 个不同形状的窗口,并利用这 9 个不同形状的窗口为每一像素视差对计算匹配代价,在这些匹配代价中选择一个最优代价作为此像素视差对的匹配代价,其数学表达式为

$$DSI_y^L(x,d)=\min_{(c_x,c_y)}\{W_y^L(x,d,w_x,w_y,c_x,c_y)\} \tag{3}$$

式中:(c_x,c_y) 为窗口中参考点的位置(图 1 中黑色方块表示可能的参考点);$W_y^L(x,d,w_x,w_y,c_x,c_y)$ 为对应窗口的匹配代价,其数学表达式为

$$W_y^L(x,d,w_x,w_y,c_x,c_y)=\sum_{s=-c_x}^{w_x-c_x}\sum_{t=-c_y}^{w_y-c_y}\| I_L(x+s,y+t)-I_R(x-d+s,y+t)\| \tag{4}$$

在支撑窗口中,为使那些与参考像素具有相同深度的像素赋予较高的贡献量,本文在多窗口的基础上加入了自适应权重函数。自适应权重函数是根据参考点与当前考虑点的空间距离与色彩距离为每一点赋予相应的权重。通过对窗口内的像素点进行分析可以得出:那些与参考点距离较近的点与参考点具有相同视差的概率大,应赋予较大的权重;同理,那些与参考点色彩相似的点与参考点具有相同视差的概率也很大,也应赋予较大的权重。根据这一原理,像素的支撑权重可以表示为考虑点与参考点之间空间距离与色彩距离的函数

$$w(p,q)=f(\Delta s_{pq},\Delta r_{pq}) \tag{5}$$

式中:Δs_{pq} 和 Δr_{pq} 分别表示像素 p、q 之间的空间距离和色彩距离。

为简化权重函数的计算,本文假定权重函数 f 可分:

$$w(p,q)=f_p(\Delta s_{pq})\cdot f_p(\Delta r_{pq}) \tag{6}$$

像素的空间支撑权重应随着与参考点空间距离的减少而增加,因此可以把函数 f_p 定义为高斯函数。函数 $f_p(\Delta s_{pq})$ 可以表示为

$$f_p(\Delta s_{pq})=\exp\left(-\frac{\Delta s_{pq}}{\gamma_s}\right) \tag{7}$$

式中:γ_s 为一个恒定常数;Δs_{pq} 为空间距离,其数学表达式为

$$\Delta s_{pq}=\sqrt{(x_p-x_q)^2+(y_p-y_q)^2}$$

同理,函数 $f_p(\Delta r_{pq})$ 可表示为

$$f_p(\Delta r_{pq})=\exp\left(-\frac{\Delta r_{pq}}{\gamma_r}\right) \tag{8}$$

式中:γ_r 为一个恒定常数;Δr_{pq} 为色彩距离,其数学表达式为

$$\Delta r_{pq}=\sqrt{(R_p-R_q)^2+(G_p-G_q)^2+(B_p-B_q)^2}$$

根据式(7)(8)，自适应权重函数(5)可以表示为

$$w(p,q)=\exp\left(-\left(\frac{\Delta s_{pq}}{\gamma_s}+\frac{\Delta r_{pq}}{\gamma_r}\right)\right) \tag{9}$$

相应的基于自适应权重的多窗口累积策略可以表示为

$$W_y^L=\frac{\sum_{s=-c_x}^{w_x-c_x}\sum_{t=-c_y}^{w_y-c_y} w(P_{\mathrm{L}}^{(x,y)},P_{\mathrm{L}}^{(x+s,y+t)})\,e(x,y,s,t,d)}{\sum_{s=-c_x}^{w_x-c_x}\sum_{t=-c_y}^{w_y-c_y} w(P_{\mathrm{L}}^{(x,y)},P_{\mathrm{L}}^{(x+s,y+t)})} \tag{10}$$

式中：$e(x,y,s,t,d)=\| I_{\mathrm{L}}(x+s,y+t)-I_{\mathrm{R}}(x-d+s,y+t)\|$。

2　动态规划立体匹配

2.1　成本函数

根据匹配序列 M，定义成本函数 $E(M)$，成本函数描述了真实对应的可能性，其数学定义为

$$E(M)=N_{\mathrm{occ}}K_{\mathrm{occ}}-N_{\mathrm{m}}K_{\mathrm{r}}+\sum_{i=1}^{N_{\mathrm{m}}}C(x_i,y_i)+\sum_{i=1}^{N_{\mathrm{m}}}\rho[\delta(x_i,y_i),\delta(x_i+1,y_i)] \tag{11}$$

式中：K_{occ} 为遮挡惩罚；K_{r} 为匹配奖励；$C(x_i,y_i)$ 为像素点 x_i 和 y_i 的匹配代价，它与相应的视差空间图 $DSI_k^L(x_i,x_i-y_i)$ 相对应；$\rho(\cdot)$ 为水平方向的平滑性约束，它是一个单调递增函数；N_{m} 和 N_{occ} 分别表示匹配序列 M 中的匹配和遮挡的像素个数。

在基于正则化的计算机视觉中，ρ 是一个二次函数，致使视差 δ 处处平滑，这导致在深度不连续处(即物体的边界处)产生较差的结果。为使成本函数在物体边界保持深度不连续，本文根据亮度梯度和视差梯度定义平滑性约束

$$\rho_d[\delta(x_i,y_i)-\delta(x_i+1,y_i)]\cdot\rho_I[\|I(x_i,y_i)-I(x_i+1,y_i)\|] \tag{12}$$

式中：ρ_I 为一个单调递减函数。

这个平滑性约束假设深度不连续与亮度边缘一致，而亮度边缘往往与物体边界重合，进而保持了边界处的视差不连续性。

2.2　动态规划限制

动态规划技术要求决策过程有序并且在任何状态的决策只取决于当前状态，为此在立体匹配中引入以下几个视差约束条件：视差范围约束，即 $0\leqslant x_i-y_i\leqslant\Delta$；次序性约束，即 $x_i\leqslant x_j\&\&y_i\leqslant y_j$；在同一匹配序列中存在遮挡匹配时，下列情况之一成立：左图像连续匹配而右图像存在遮挡，或者右图像连续匹配而左图像存在遮挡，即 $x_{i+1}=x_i+1\,\|\,y_{i+1}=y_i+1$。假设匹配序列中当前匹配为 (x_i,y_i)，根据上述约束，其后继匹配可分为 3 种情况：① 对角匹配，即 (x_i+1,y_i+1)；② 垂直跳跃，即 $(x_i+1,y_i+2),(x_i+1,y_i+3),\cdots$；③ 水平跳跃，即 $(x_i+2,y_i+1),(x_i+3,y_i+1)\cdots$。具体过程如图 2 所示。

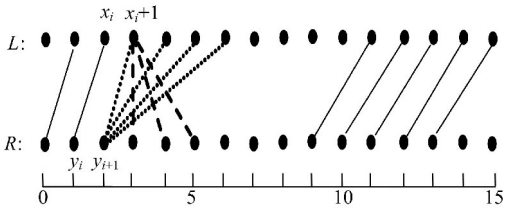


图 2　匹配状态选择

Fig. 2　Choice of match state

通过以上分析，在匹配网格中对于每一匹配 (x_i,y_i) 其直接前驱匹配如图 3(a)所示，其直接后继匹配如图 3(b)所示。图 3 显示了对于每一匹配都有 $\Delta+1$ 个候选直接前驱匹配和 $\Delta+1$ 个候选直接后继匹配，其中 Δ 为视差范围。对于格网中的每一单元，使用 $\varphi(x_i,y_i)$ 记录匹配序列到达 (x_i,y_i) 点时的最优匹配成本，式(11)表示的成本函数可转化为

$$\varphi(x_i,y_i)=C(x_i,y_i)-K_r+\min\begin{cases}\varphi(x_i-1,y_i-1)\\ \varphi(x_i-2,y_i-1)+K_{\mathrm{occ}}+E_{\mathrm{s}},\cdots,\\ \varphi(x_i-k,y_i-1)+K_{\mathrm{occ}}+E_{\mathrm{s}}\\ \varphi(x_i-1,y_i-2)+K_{\mathrm{occ}}+E_{\mathrm{s}},\cdots,\\ \varphi(x_i-1,y_i-k)+K_{\mathrm{occ}}+E_{\mathrm{s}}\end{cases} \tag{13}$$

式中：E_{s} 为式(12)所表达的平滑成本。

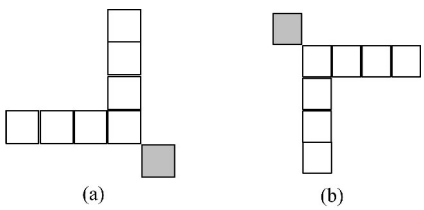


图 3　前驱及后继匹配

Fig. 3　Preceding and following match

2.3　限制搜索空间

为减少基于动态规划立体匹配算法的搜索空间，以提高匹配速度，本文对搜索空间进行了限

制。首先通过快速立体匹配算法获得初始匹配序列,并利用初始匹配序列对搜索空间进行限制,使其在初始匹配序列的邻域内搜索,这样可以很大程度地减少搜索空间中的状态数,提高算法速度。

假设初始匹配序列为{(1,0),(2,1),(3,2),(5,3),(6,4),(7,5),(8,7),(9,8)},如图 4 所示,这个匹配序列在图 4 中已使用符号"×"标记。在传统的动态规划立体匹配中,搜索空间是一个大小为 N^2 的二维数组,其中 N 是图像的宽度。由于相机所拍摄的景物离相机都有一定的距离,因此,所有像点的视差值都在一定范围内,这样可以使搜索空间进一步减少为 $N\times\Delta$。虽然通过视差范围约束可以减少容许访问的搜索空间数量,但对于在宽基线下形成的立体像对而言,视差范围较大,导致搜索空间依然很大。为此,本文在进行动态规划之前引入一个初始匹配序列(如图 4 所示),在初始匹配的周围邻域内选择一个宽度为 W 的"条形带"作为动态规划的搜索空间,这样可以使搜索空间数量减少为 $N\times W$,其中 W 为"条形带"的宽度。一般情况下,"条形带"的宽度要远小于视差范围 Δ,因此,可以通过此方法使搜索空间的可访问单元减少 $N\times(\Delta-W)$ 个。

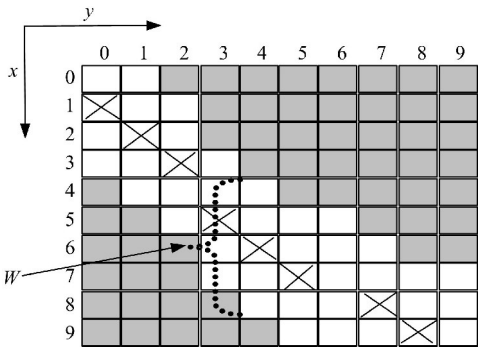


图 4 搜索空间限制

Fig. 4 Limitation of search space

3 实验分析

在 Windows XP 下使用 VC++6.0 实现了本文算法。实验所用的计算机为双核 CPU2.2 GHz,内存 1 GB。下面做了 3 组实验对本算法进行验证,在第一、二组实验中采用立体像对 Tsukuba 分析了"条形带"宽度和初始匹配序列对匹配的影响,Tsukuba 图像对中含有较多对象,场景复杂,图像的视差变化范围较大,这幅立体像对的最大视差范围为 30,实验使用的立体像对和真实视差图如图 5 所示。第三组实验采用多个场景分析验证了本文算法的匹配精度和运行效率。

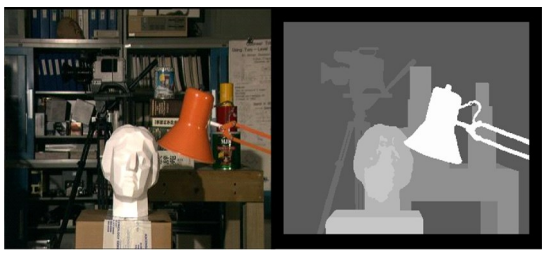
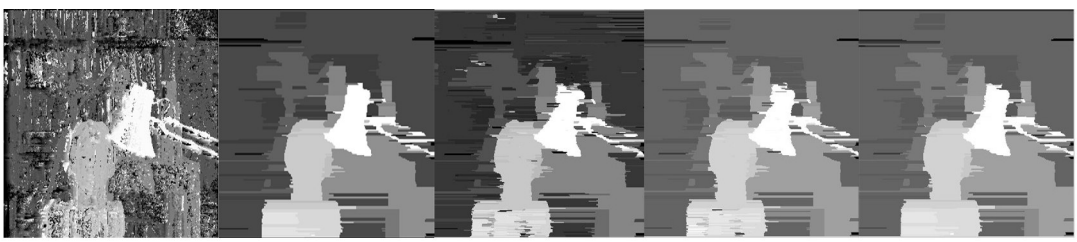


图 5 Tsukuba 立体像对

Fig. 5 Tsukuba stereo images

3.1 空间限制分析

本实验分析验证了"条形带"宽度 W 对匹配结果的影响。本文提出的算法是以初始视差图为基础对搜索空间进行限制,再利用动态规划求解最终视差图。在匹配过程中,首先利用当前行的初始视差值形成初始匹配序列,再利用此序列和"条形带"宽度 W 对搜索空间进行限制。当 $W=0$ 时,搜索空间为初始匹配序列构成的一条曲线,在此基础上运用动态规划求解获得的视差图没有发生变化,即为原视差图。随着 W 的增加,搜索空间也随着相应增加,匹配精度和匹配时间也发生相应的改变。图 6(a)为实验的初始视差图,其



(a) 初始解 (b) 非限制解 (c) W=10 (d) W=15 (e) W=20

图 6 空间限制分析

Fig. 6 Analysis of limited search space

坏点百分比为 49.95%；(b)显示了未限制搜索空间的视差图，其坏点百分比为 10.78%；(c)显示了在 $W=10$ 的搜索空间中进行匹配获得的结果，其坏点百分比为 18.46%；(d)显示了 $W=15$ 的测试结果，其坏点百分比为 13.2%；(e)显示了 $W=20$ 的测试结果，其坏点百分比等于非限制搜索空间的坏点百分比。实验证明：通过调节“条形带”宽度可以获得精度较高的视差图，而此时每个扫描行可以减少 3840 个搜索单元，整个匹配过程可以减少 288×3840 个搜索单元。

3.2　初始解分析

本实验分析验证了初始匹配序列对匹配结果的影响。本文提出的立体匹配算法是在初始匹配序列的邻域内应用动态规划算法求解视差图，其解的好坏与其邻域内是否存在一个全局最优解路径有关，如果在其邻域内存在全局最优解，则算法可以获得其最优解；如果最优路径不在其邻域内，本算法将获得一个次最优解路径。本实验在“条形带”宽度 $W=10$ 的情况下，以不同精度的视差图构造初始匹配序列进行实验，测试对比结果如图 7 所示，从实验结果可以看出，初始匹配精度相对较好时即可获得精度较高的视差图。

3.3　匹配精度及时间分析

本实验对本文算法的匹配精度和运行时间进行了验证，采用的立体像对如图 8(a)(b)所示，自上而下分别为 Cones，Teddy，Venus，实验结果如图 8(d)所示。对于实验获得的视差图，采用了坏点百分比对其进行评价：

$$P=\frac{1}{N}\sum\left(\left|d_{\mathrm{c}}(x,y)-d_{\mathrm{T}}(x,y)\right|>\delta_{\mathrm{d}}\right)$$

式中：$d_{\mathrm{c}}(x,y)$ 为计算视差图；$d_{\mathrm{T}}(x,y)$ 为真实视差图；δ_{d} 为视差错误阈值；N 为图像的像素总数。

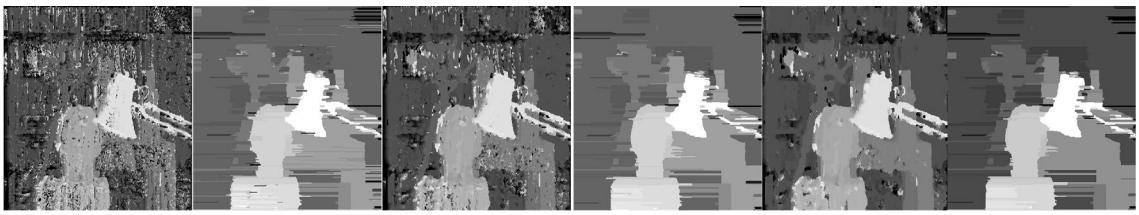


图 7　初始解分析

Fig. 7　Analysis of initial solution

本文采用 $\delta_{\mathrm{d}}=1$ 对本文算法和当前几个匹配效果最好的算法进行比较，其性能如表 1 所示。表 1 中的每一列分别显示不同算法作用于同一立体像对的坏点百分比，值越小，表明该算法匹配效果越好。由表 1 可见，本文算法的性能不仅优于局部立体匹配算法[2]及 DP 算法[6-7]，而且还优于全局的 graph cut 算法[9]。从运行时间的角度上看，由于算法限制了搜索空间，使匹配速度有了明显的改进，如表 2 所示，它显示了不同立体像对在限制搜索空间和非限制搜索空间下的运行时间。表 3 显示了本文算法与其他几种算法的运行时间。由表 3 可见，基于局部的和 DP 算法运行时间最快，基于全局的算法运行时间最慢，而本文算法的运行时间优于这些算法。

实验结果表明，本文算法是一种高效可靠的立体匹配算法，它不但可以获得精度较高的视差图，而且显著改善了算法的实时性。

表 1　不同算法的误差比较

Table 1　Comparison of error rate among different algorithms　　%

算法	测试图		
	Cones	Teddy	Venus
BP[10]	5.96	4.48	1.96
本文算法	7.37	5.27	2.18
Graph Cut[9]	8.25	7.13	3.52
文献[7]算法	10.56	9.30	4.43
文献[6]算法	12.28	10.28	4.37
文献[2]算法	14.70	12.80	5.30

表 2　限制与非限制搜索空间的时间对比

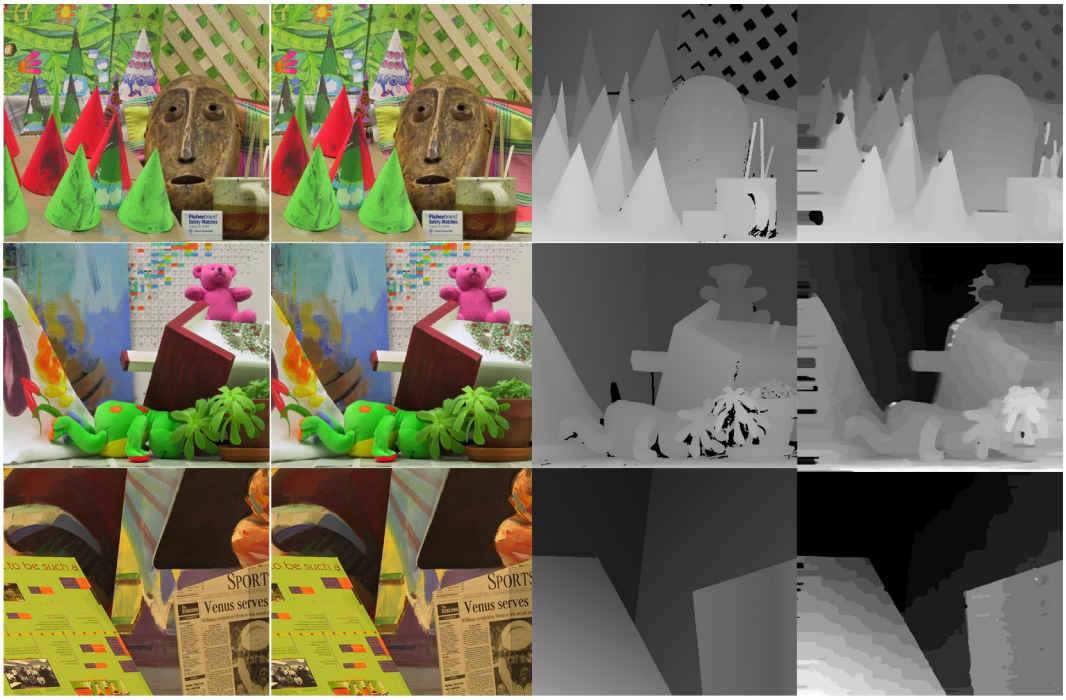
Table 2　Comparison of running time for limited and unlimited search space

立体像对	分辨率	最大视差	非限制时间/s	限制时间/s
Cones	450×375	60	15.30	4.73
Teddy	450×375	45	10.90	3.64
Venus	434×383	30	8.10	2.04

表 3　不同算法的时间对比

Table 3　Comparison of running time among different algorithms

算法	运行时间/s		
	Cones	Teddy	Venus
本文算法	4.73	3.64	2.04



(a) 左图像 (b) 右图像 (c) 真实视差图 (d) 计算视差图

图 8 实验结果

Fig. 8 Experimental results

文献[7]算法	20.56	15.30	7.43
文献[2]算法	107.35	103.14	68.57
BP[10]	183.60	167.50	145.20
Graph Cut[9]	285.58	231.74	205.20

4 结束语

提出的立体匹配算法通过初始匹配序列对搜索空间进行限制,再通过"条形带"宽度的调节很大程度上减少了搜索空间的规模。通过在平滑函数中引入亮度梯度限制有效地避免了平滑函数对物体边界处的视差变化产生过度惩罚。通过基于自适应权重的多窗口累积策略提高了匹配精度。实验验证了该立体匹配算法不但可以获得高精度的视差图,而且在此基础上提高了匹配速度。

参考文献:

[1] Scharstein D, Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms[J]. International Journal of Computer Vision, 2002, 47(1-3):7-42.

[2] Yoon K J, Kweon S. Adaptive support-weight approach for correspondence search[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(4): 650-656.

[3] De-maeztu L, Villanueva A, Cabeza R. Stereo matching using gradient similarity and locally adaptive support-weight[J]. Pattern Recognition Letters, 2011, 32(13): 1643-1651.

[4] Geiger D, Ladendorf B, Yuille A. Occlusions and binocular stereo[J]. International Journal of Computer Vision, 1995, 14(3):211-226.

[5] Cox I J, Hingorani S L, Rao S B, et al. A maximum likelihood stereo algorithm[J]. Computer Vision Image Understanding, 1996, 63(3):542-567.

[6] Birchfield S, Tomasi C. Depth discontinuities by pixel-to-pixel stereo[J]. International Journal of Computer Vision, 1999, 35(3):269-293.

[7] Bobick A F, Intille S S. Large occlusions stereo[J]. International Journal of Computer Vision, 1999, 33(3):181-200.

[8] 刘赫伟,汪增福.一种沿区域边界的动态规划立体匹配算法[J].模式识别与人工智能,2010,23(1):38-44.
Liu He-wei, Wang Zeng-fu. Stereo matching algorithm based on dynamic programming along region boundary[J]. Pattern Recognition and Artificial Intelligence, 2010, 23(1): 38-44.

[9] Boykov Y, Veksler O, Zabih R. Fast approximate energy minimization via graph cuts[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(11): 1222-1239.

[10] Sun J, Zheng N N, Shum H Y. Stereo matching u-

sing belief propagation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(7): 787-800.