

文章编号: 0253-2239(2009)04-1002-08

# 一种基于分割的可变权值和视差估计的立体匹配算法

刘天亮 罗立民

(东南大学 影像科学与技术实验室, 江苏 南京 210096)

**摘要** 立体匹配一直是计算机视觉研究领域中的热点和难点,是立体视觉中的关键技术之一。为了消除基于局部图像的双目立体匹配的歧义性,提出一种基于图像分割及可变权值方案的初始匹配和贪婪的后处理视差估计策略相结合的立体匹配算法。分割彩色立体图像对,利用分割自适应地分配权值来消除匹配特征相似的歧义性,计算匹配代价得到初始视差。接着,为了更好地消除弱纹理区域、重复纹理区域和宽遮挡区域等复杂歧义性,视差后处理中采用贪婪估计方案,包括基于分割的视差校准、窄遮挡处理及多方向自适应加权最小二乘拟合填充。实验结果表明,基于分割的本算法结构简单,能有效地提高处理局外点的稳健性,并生成高精度的稠密视差。

**关键词** 机器视觉; 双目立体匹配; 可变权值; 分割; 视差校准; 加权最小二乘

**中图分类号** TP391.41 **文献标识码** A **doi**: 10.3788/AOS20092907.1002

## Segmentation-Based Stereo Matching Algorithm with Variable Support and Disparity Estimation

Liu Tianliang Luo Limin

(Laboratory of Image Science & Technology, Southeast University, Nanjing, Jiangsu 210096, China)

**Abstract** Stereo matching is a long active topic and difficult problem in computer vision, and is a crucial technique in stereovision. An algorithm by combining initial matching via segmentation-based variable support with greedy disparity estimation as post-processing is proposed to resolve the ambiguity of binocular stereo problem in a local perspective. Firstly, color segmentation is conducted on both stereo images, and segmentation-based adaptive support weight is assigned for each pixel to eliminate ambiguity in feature matching, and then matching cost with the variable support is calculated to obtain initial disparity. Secondly, to address more other complex ambiguity in low textured and repetitive patterns or large occluded regions etc., greedy disparity estimation procedure consists sequentially of three steps: segmentation-based disparity calibration, narrow occlusion handling and multi-directional weighted least square fitting. The experimental results indicate that this technique with segmentation cues can increase robustness against outliers and obtain accurate and dense disparity effectively. It's concise and efficient.

**Key words** machine vision; binocular stereo matching; variable support; segmentation; disparity calibration; weighted least square

## 1 引 言

立体匹配中双目视觉,即仅处理经极线校正后的两张立体图像;精确且稠密地立体匹配,在三维表面建模、机器人导航、工业监控及基于图像的绘制等

方面得到了广泛应用<sup>[1]</sup>。文献[2]很好地总结了目前常见的一些立体匹配算法及其评判标准。总体来说,立体匹配算法大致可分为如下两类:基于全局<sup>[1,3]</sup>和局部图像<sup>[4]</sup>的算法。近来文献[5]折衷于上

收稿日期: 2008-06-25; 收到修改稿日期: 2008-09-26

作者简介: 刘天亮(1980—),男,博士研究生,主要从事计算机视觉、图像处理和模式识别等方面的研究。

E-mail: ltl315@gmail.com

导师简介: 罗立民(1956—),男,教授,博导,主要从事医学影像处理、科学可视化和立体定向放射治疗等方面的研究。

E-mail: luo.list@seu.edu.cn

述两类,提出了基于半全局的算法。基于全局及半全局的算法,一般能获得精度较高的视差图,但相应的参数设置较难且复杂度也较高。而基于局部的算法,能很快且较容易地恢复出高纹理区域的视差。

自适应权值(Adaptive support-weight, ASW)算法是近来双目视觉中一种有效的窗口选择算法<sup>[6]</sup>。ASW 算法是以待匹配点为中心选取一个固定的支持窗,同时利用色彩相似性测度和空间距离远近等信息分配窗口内的各点与待匹配点之间的相应权重。为构造可靠的不相似性测度,可利用 ASW 算法获得可靠的匹配代价<sup>[6]</sup>;为获得精确且平滑的视差,可以基于 ASW 算法来校准视差<sup>[7,8]</sup>。但 ASW 算法考虑了空间距离因素,易导致视差不连续、低纹理及重复纹理区域出现误匹配<sup>[9]</sup>。立体问题中一般存在视差间断、遮挡等,也可能有较复杂的宽遮挡。窄遮挡处理已有较好的算法<sup>[10]</sup>;但对较宽遮挡,该算法效果欠佳,有待改进。

为了解决基于局部图像的歧义性问题,本文统一地利用分割将基于可变权重化的初始匹配和贪婪的视差后处理有机结合来完成立体匹配。实验结果表明,对消除图像歧义性,本匹配算法取得了较为满意的结果。

## 2 基于分割的立体匹配算法概述

匹配算法可分两步骤:初始匹配和视差估计(如图 1 所示)。首先,对立体图像进行色彩分割,利用分割的自适应权重化(Segmentation-based adaptive support-weight, SASW)算法构造匹配代价,得到稠密的初始视差图。然后,基于贪婪思想,利用 SASW 算法将初始视差校准成平滑、无噪声且有精

确视差边界的视差图;基于对称性视差一致性约束检测遮挡,利用相同分割且纹理上最接近或最相似假设填充窄遮挡等区域;多方向地自适应加权最小二乘拟合,填充宽遮挡等复杂歧义区域。

## 3 初始匹配

### 3.1 图像分割和初始匹配代价

对立体图像都进行分割,并假设每个分割内的像素具有相近的视差值。本算法采用基于 Mean-shift 的高效图像分割算法<sup>[11]</sup>。该分割算法是一种利用概率分布的梯度寻找分布峰值的稳健性高的非参数估计方法,能将图像中每一个像素归类到相应的密度模式下,从而实现聚类并得到一系列互不交叉的区域;具有精度高和实时性较好等特点,且适当调整参数在 CIELab 色彩空间中能对图像进行过分割。如果分割区域过大,则该分割内的视差可能不大相近。为较安全地满足每个分割内视差平滑变化假设,应该对彩色图像强过分割形成够小区域块<sup>[1]</sup>。

基准图像(Base image)像素  $p$  和关于视差  $d$  的待匹配图像(Matching image)像素  $q_d$  的匹配程度,可简单地用  $p$  和  $q_d$  之间色彩差绝对值和某一阈值( $T_{AD}$ )来衡量<sup>[4]</sup>。则初始匹配代价函数  $e(p, q_d)$  公式表达如下:

$$e(p, q_d) = \min \left\{ \sum_{c \in \{r, g, b\}} |I_c(p) - I_c(q_d)|, T_{AD} \right\}, \quad (1)$$

式中  $I_c$  为色彩分量  $c$  的强度分布;根据图像强度分布的先验,  $T_{AD}$  可以限制匹配成本范围,以降低误匹配。

### 3.2 基于 SASW 的匹配计算

SASW 算法是双目视觉中一种非常有效且新颖的窗口选择算法<sup>[9]</sup>。该 SASW 算法启发于自适应权值 ASW 算法<sup>[6]</sup>,引入了色彩分割一致性约束,在逐点选取的支持窗内,仅根据即得的色彩分割和邻近的色彩相似性分配各点与待匹配点之间的相应权重,从而获得最可靠的匹配代价。

基于区域匹配的基本思想是,如果左右图像中两点是正确的匹配,那么其邻域的对应该像素也是正确的匹配,所以在匹配过程中,通常以邻域相似性作为测度。假设  $I_b$  和  $I_m$  分别为基准图像和待匹配图像。为计算  $I_b$  内某点  $p_b$  和待匹配点  $q_{m,d} \in I_m$  之间的匹配成本,  $I_b$  内每一点对应的权值不仅取决于与  $p_b$  的色彩差异,同时也取决于上述分割处理得到的

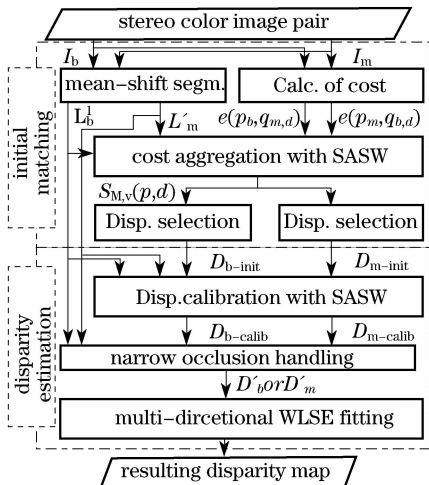


图 1 算法流程图

Fig. 1 Flowchart of the proposed algorithm

分割区域的信息。则  $I_b$  中  $p_b$  对邻域点  $p_i$  的权值可定义为

$$\omega_b(p_i, p_b) = \begin{cases} 1.0 & p_i \in S_b \\ \exp\left(-\frac{d_c(I_b(p_i), I_b(p_b))}{\gamma_c}\right) & \text{otherwise} \end{cases} \quad (2)$$

式中  $S_b$  是  $p_b$  点所在的分割区域,  $d_c$  为  $p_b$  与  $p_i$  之间的色彩差异, 而常数  $\gamma_c$  是本算法的经验值。从(2)式可知, 对于分割  $S_b$  之外所有像素的权值, 未考虑距离差异的影响; 而且相关窗口内所有像素的权值, 也独立于与中心点  $p_b$  的距离。前期分割在 CIELab 色彩空间, 因此增强了支持权重的稳健性; 为计算简便, 判断不属于当前点分割的色彩相似度, 不是在 CIELab 色彩空间中度量<sup>[6,12]</sup>, 而是直接将其放在 RGB 空间进行衡量, 则色彩差异为

$$d_c[I_b(p_i), I_b(p_b)] = \sqrt{\sum_{c \in \{r, g, b\}} |I_{b,c}(p_i) - I_{b,c}(p_b)|^2}, \quad (3)$$

式中  $I_{b,c}$  为  $I_b$  的色彩分量  $c$  的强度。同理, 对  $I_m$  中  $q_{m,d}$  对邻域点  $q_d$  的权值, 可进行类似计算。对每一对匹配点的相似测度中, 本策略将在基准图像和待匹配图像中分别地计算两个不同的支持权重。

一旦上述权重计算后, 某点  $p_b$  和待匹配点  $q_{m,d}$  之间的匹配代价, 则可采用对称策略进行计算; 即对两图像点  $p_b$  和  $q_{m,d}$  在匹配窗口内的初始匹配成本与相应权值进行点乘积, 并规一化于相应权值本身的点乘积:

$$S_{M,v}(p_b, q_{m,d}) = \frac{\sum_{p_i \in Np_b, q_d \in Nq_{m,d}} \omega_b(p_i, p_b) \cdot \omega_m(q_d, q_{m,d}) \cdot e(p_i, q_d)}{\sum_{p_i \in Np_b, q_d \in Nq_{m,d}} \omega_b(p_i, p_b) \cdot \omega_m(q_d, q_{m,d})}, \quad (4)$$

式中  $q_d$  和  $q_{m,d}$  是在视差为  $d$  的情况下,  $I_b$  中像素点  $p_i, p_b$  在匹配图像  $I_m$  中的对应点。  $Np_b$  和  $Nq_{m,d}$  分别为  $p_b$  和  $q_{m,d}$  点的矩形支持窗口,  $e(p_i, q_d)$  是初始匹配成本。

采用优胜者全选 (Winner takes all, WTA) 作为视差选择方法。设  $d_{\max}$  和  $d_{\min}$  分别是视差最大值和最小值, 则预定的视差范围  $R_d = [d_{\min}, d_{\max}]$ , 且搜索范围中的视差个数  $N_d$  为  $d_{\max} - d_{\min} + 1$ 。则  $I_b$  中的初始匹配所得到的视差  $D_{b-\text{init}}(p_b)$  为

$$D_{b-\text{init}}(p_b) = \arg \min_{d \in R_d} S_{M,v}(p_b, q_{m,d}) \quad (5)$$

同理, 对  $I_m$  中的初始视差  $D_{m-\text{init}}(q_{m,d})$ , 可进行

类似计算。

### 3.3 基于 SASW 的初始匹配

基于 SASW 的初始匹配算法的具体流程:

初始化: 对初始视差  $D_{b-\text{init}}(p_b)$  和  $D_{m-\text{init}}(q_{m,d})$ , 设置初值为零, 并转第一步。

第一步: 分割彩色立体图像对, 并根据(1)式计算图像  $I_b, I_m$  中各自初始匹配代价  $e(p_i, q_d)$ ;

第二步: 利用即得的分割基于 SASW 算法, 对图像  $I_b, I_m$  中每一像素点在搜索范围  $R_d$  内每一个视差值  $d$ , 根据(2)、(3)及(4)式对称地计算相应的匹配代价;

第三步: 图像  $I_b$  中所有点  $p_b$ , 根据(5)式, 计算  $I_b$  的初始视差  $D_{b-\text{init}}(p_b)$ ; 同理, 可得  $I_m$  的初始视差  $D_{m-\text{init}}(q_{m,d})$ , 此时初始匹配算法结束。

## 4 视差估计

### 4.1 基于 SASW 的视差校准

理想情况下, 在一定区域内颜色相同的点具有相似的视差值。因此, 可以根据某点周边一定区域内所有点的初始视差分布, 来校准该点的视差值。对待校准点  $p_c$ , 首先为其确定一个可靠的校准窗口, 使得窗口内尽可能只包含同一深度的像素点; 然后在预定的视差搜索范围内, 统计地计算各个视差值在校准窗内出现的频率 (即视差分布情况), 将频率最高的视差值选为  $p_c$  点的最终视差值。

校准窗的选取和匹配代价的计算类似, 以待校准点为中心选取一个大小固定的窗, 然后根据窗内各点与待校准点的色彩相似程度来分配相应的权重。将窗的选择问题转化为固定大小的窗内各点的权重分配问题。为了便于计算, 且尽可能地使校准窗内只包含相同深度的像素点, 本算法利用单目图像内的分割信息, 自适应地分配权重, 并以此校准视差。即仅考虑当前视图中待校准点邻域内的权重。

现以基准图像  $I_b$  为例。假设  $I_b$  中一待校准点  $p_c$ , 其初始匹配得到的视差为  $D_{\text{init}}(p_c)$ ,  $p_i$  是  $I_b$  中以  $p_c$  为中心的校准窗口内的邻近点, 则  $p_i$  和  $p_c$  之间的权重类似于匹配代价中相应的权重, 且也未曾考虑距离因素, 即公式描述为

$$\omega_b(p_i, p_c) = \begin{cases} 1.0 & p_i \in S_c \\ \exp\left(-\frac{d_c(I_b(p_i), I_b(p_c))}{\gamma_b}\right) & \text{otherwise} \end{cases} \quad (6)$$

与(2)式虽然相似, 但(6)式也有不同, 其中常数  $\gamma_b$  是校准用的经验值。由于视差校准中假设校准

窗内所有的像素点有相同的色彩,此处分配的可变权重  $\omega_h(p_i, p_c)$ , 在不同于当前点分割  $S_c$  的色彩相似性上, 色彩应具有更加严格的约束。为防止  $\omega_h(p_i, p_c)$  敏感于色彩变化,  $\gamma_h$  应该小于或等于  $\gamma_c$ , 而且校准窗口的尺寸也可以适当地考虑小于或等于匹配窗口的尺寸(但也不能过小)。

为了简洁地描述该校准过程, 首先定义一个  $u \times v \times N_d$  的三维矩阵  $\mathbf{M}$ , 并初始化为零值( $u$  和  $v$  是基准图像的行数和列数)。然后根据初始视差  $D_{b\text{-init}}(p_c)$  的值对  $\mathbf{M}$  矩阵进行赋值:

$$\mathbf{M}[p_{cx}, p_{cy}, D_{b\text{-init}}(p_c) - d_{\min} + 1] = 1, \quad (7)$$

上式表示, 对  $I_b$  中  $(p_{cx}, p_{cy})$  为坐标且  $D_{b\text{-init}}(p_c)$  为初始视差的某一点  $p_c$ , 将  $\mathbf{M}$  矩阵中第  $p_{cx}$  行  $p_{cy}$  列的  $N_d$  个视差级中, 关于  $D_{b\text{-init}}(p_c)$  视差值的第  $D_{b\text{-init}}(p_c) - d_{\min} + 1$  个视差级相应的矩阵位置值赋为 1, 其他位置仍置为零。对  $I_b$  中每一个像素点, 利用(7)式遍历一次, 就可以获得完全对应于  $D_{b\text{-init}}(p_c)$  的一个三维矩阵  $\mathbf{M}$ 。最后, 对于待校准点  $p_c$ , 在视差范围  $R_d$  内关于每一视差切片  $d$ , 利用(6)式得到的相应邻域权重与(7)式得到的三维矩阵  $\mathbf{M}$  进行点乘积, 并基于每一视差  $d$  点乘积的求和值进行最大化, 选取最大值所对应的视差为校准得到的视差, 以此实现视差校准的计算过程, 即公式表达如下:

$$D_{b\text{-calib}}(p_c) = \arg \max_{d \in R_d} \left\{ \sum_{p_i \in Np_c} \omega_h(p_i, p_c) \cdot M(p_{ix}, p_{iy}, d - d_{\min} + 1) \right\}, \quad (8)$$

式中  $Np_c$  是以  $p_c$  点为中心且水平和垂直方向上的校准窗宽分别为  $W_{cx}$ 、 $W_{cy}$  的矩形支持窗, 也即  $Np_c =$

$$\left\{ (x, y) \left| \begin{array}{l} x_c - \frac{W_{cx}}{2} \leq x \leq x_c + \frac{W_{cx}}{2} \\ y_c - \frac{W_{cy}}{2} \leq y \leq y_c + \frac{W_{cy}}{2} \end{array} \right. , p_c = (x_c, y_c) \right\}.$$

$D_{b\text{-calib}}(p_c)$  为校准之后的视差值。

视差校准的目标是获得足够平滑、不包含噪声且保持精确的视差边界的视差图。若一次校准的效果不佳, 可以利用前一次校准得到的视差图, 贪婪地进行二次或多次校准。校准待匹配图像  $I_m$  的视差  $D_{m\text{-calib}}(q_{m,d})$ , 可进行类似计算。

#### 4.2 窄遮挡处理

在讨论宽遮挡等复杂歧义之前, 对校准后的平滑视差图, 先做窄遮挡处理。不可靠视差检测及简单地贪婪视差填充, 是双目视觉中一种新颖且效果较好的窄遮挡处理算法<sup>[10]</sup>。本文主要讨论宽遮挡

等, 此处窄遮挡方法基本类似于文献[10]中部分算法。对窄遮挡方法, 现可归纳并总结为以下几步骤(以基准图像  $I_b$  为例):

第一步: 依据同一场景在两个视图之间形成的视差限制在阈值  $T_{occ}$  内的一致性约束准则, 对称地检测遮挡区域。

第二步: 利用即得图像分割区域, 对互不交叉的每一分割, 根据相同分割内总遮挡视差数目、每一视差聚类中心的数目与区域面积的比值  $R_{ro}$ 、 $R_{rc}$  及相应的阈值  $T_{ro}$ 、 $T_{rc}$  相结合, 来剔除相同分割内不可靠的视差。该剔除方案是, 若  $R_{ro} > T_{ro}$ , 则剔除分割内所有视差; 否则, 再判断每一视差聚类的  $R_{rc} \leq T_{rc}$  是否成立, 如果条件成立, 则该类视差也被剔除。

第三步: 对每一个待填充像素  $p(x, y)$ , 依次借助双目和单目的信息, 根据色彩差异排序贪婪地进行搜索且每次均起始于更新的最小排序位置, 将八邻域内属于相同分割且亮度差异  $D_g$  小于阈值  $T_s$  的可靠视差, 赋值给当前  $p(x, y)$ , 结束当前搜索并转下一个不可靠视差, 如此反复地实现窄遮挡等不可靠区域填充。利用双目信息时,  $D_g = |I_b(x, y) - I_m(x-d, y)|$ ,  $d$  是当前搜索的邻域像素的视差值; 而基于单目信息时,  $D_g = |I_b(x, y) - I_b(x+m, y+n)|$ ,  $m$  和  $n$  仅限于  $-1, 0, +1$ , 但  $(m, n) \neq (0, 0)$ 。

第四步: 用中值滤波对视差再做平滑处理。

#### 4.3 多方向自适应加权最小二乘拟合填充

本节提出基于多方向自适应加权最小二乘拟合策略, 填充宽遮挡等复杂且依然不可靠的区域(以基准图像  $I_b$  为例)。以上窄遮挡处理后, 仍可能有较大的不可靠区域。初始匹配不能保证每一不可靠视差邻域内至少有一个可靠视差满足窄遮挡处理中的填充条件。不同于以往的只对亮度本身且仅限于水平方向内一侧或两侧的拟合策略<sup>[7]</sup>, 忽略彩色分割区域边界, 充分地利用以当前不可靠点  $p_b$  为起始点的  $K$  个角度分布均匀的径向路径上丰富的单目信息(如彩色、灰度及几何信息)。  $K$  取值范围为  $2 < K \leq 36$ 。凸包内为未分配视差, 而凸包本身及外面是经过上述处理已分配视差(图 2)。在理想情况下, 立体场景中激烈的亮度变化, 往往伴随着相应视差发生巨大的变化; 且凸包内视差值, 限定在凸包边上像素位置的视差范围。填充策略是, 将其所有方向中最小亮度变化的那个方向上的视差, 贪婪地分配给当前待填充点  $p_b$ , 即  $p_b$  点视差值一经分配后, 就不再改动。

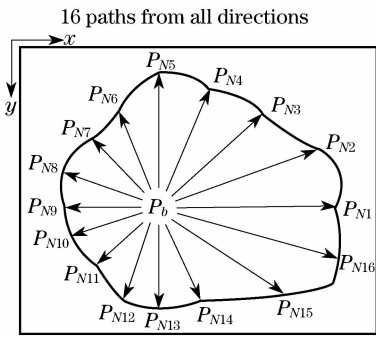


图 2 所有方向加权最小二乘路径示意图  
Fig. 2 Paths in all directions for weighted  
least-square-errors fitting

填充时需先对起始于  $p_b$  终止于每个 1D 径向上第一个已分配视差  $p_{Nk}$  的路径上灰度分布, 计算自适应加权最小二乘误差值 (Weighted Least Square Errors, WLSE)。该 WLSE 值反映该路径上关于色彩和几何信息的权重化亮度变化情况。设图像  $I_b$  的灰度分布为  $I_{b,L}$ 。对所有方向上测度值最小化, 则最小值路径上对应  $p_{Nk}$  的视差值为  $p_b$  点视差, 表达如下:

$$p_b^* = \arg \min_{k=1, \dots, K} \{f_1(p_{N1}), \dots, f_k(p_{Nk}), \dots, f_K(p_{NK})\}, \quad (9)$$

其中

$$f_k(p_{Nk}) = \left\{ \sum_{i=1}^{Nk} \omega(p_b, p_{ki}) * [I_{b,L}(p_{ki}) - \text{Mean}(p_b, k)]^2 \right\} / \sum_{i=1}^{Nk} \omega(p_b, p_{ki}), \quad (10a)$$

$$\text{Mean}(p_b, k) = \frac{1}{Nk} \sum_{i=1}^{Nk} I_{b,L}(p_{ki}), \quad (10b)$$

$$\omega(p_b, p_{ki}) = \exp \left\{ - \left[ \frac{\Delta C_{p_b, p_{ki}}}{\gamma_{fc}} + \frac{\Delta G_{p_b, p_{ki}}}{\gamma_{fg}} \right] \right\}, \quad (10c)$$

$$\Delta C_{p_b, p_{ki}} = \sqrt{\sum_{c \in \{L, a, b\}} [I_{b,c}(p_b) - I_{b,c}(p_{ki})]^2}, \quad (10d)$$

$$\Delta G_{p_b, p_{ki}} = \| p_b - p_{ki} \|_2, \quad (10e)$$

(10a)式和(10b)式分别代表  $p_b$  点所在的第  $k$  个径向路径上的权重化亮度变化和对应的均值;  $p_{ki}$  是长度为  $Nk$  的该路径上第  $i$  个不可靠像素。(10c)式表示  $p_b$  和  $p_{ki}$  关于色彩和几何的自适应支持权重<sup>[6]</sup>, 式中  $\gamma_{fc}$  和  $\gamma_{fg}$  分别为色彩差异和几何距离的权重系数; (10d)式和(10e)式分别为  $p_b$  和  $p_{ki}$  在 CIELab 空间的色彩差异及几何空间的距离。各路径上的相应权值, 考虑了几何距离因素; 则最近邻的有效视差, 在填充时能有效地将其凸现并用于填充。

$f_k(p_{Nk})$ , 实际上是以给定路径上  $p_b$  和  $p_{Nk}$  之间从亮度、色彩及图像几何等线索来构造的感知距离; 作为处理复杂遮挡等视差填充准则, 将是一个非常有效的测度。

基于多方向权重化的宽遮挡填充算法, 可分为以下几步骤:

第一步: 对即得的视差图, 逐点搜索仍未赋予可靠视差的点  $p_b$ , 并转第二步;

第二步: 对起始于  $p_b$  的每个 1D 径向路径, 根据(10a-10e)各式计算  $f_k(p_{Nk})$ ;

第三步: 对  $p_b$  的  $K$  个径向路径上视差值  $p_{Nk}$  和 WLSE 值  $f_k(p_{Nk})$ , 根据(9)式择优选取  $p_{Nk}$  作为  $p_b$  的视差值, 并转第一步, 搜索下一个待填充点位置;

第四步: 填充完毕所有的不可靠视差点, 用中值滤波视差, 获得最终视差图。

## 5 实验结果及分析

用本文算法分别对 Middlebury 网站<sup>[2]</sup>提供的 4 对彩色测试图像 Tsukuba、Venus、Teddy 和 Cones 进行了匹配并对结果做正确匹配率的统计分析, 以此验证本算法的有效性。对 4 个立体图像对的计算过程中, 按照测试要求, 本算法都保持相同参量。初始匹配中 Mean-shift 彩色分割中几何空间窗和特征空间窗的半径均设为 3 且允许的最小区域为 35 像素,  $T_{AD} = 50$ , 匹配窗口为  $33 \times 33$  且  $\gamma_c = 20$ ; 视差估计中校准窗口为  $25 \times 25$  且  $\gamma_h = 12$ , 校准迭代为 1 次,  $T_{acc} = 2$ ,  $T_m = 0.75$ ,  $T_r = 0.05$ ,  $T_s = 4$ ,  $K = 36$ ,  $\gamma_{fc} = 5$  及  $\gamma_{fg} = 17.5$ ; 算法评估的误匹配阈值设为 1。

图 3 为本算法处理 4 对测试图像得到的分割结果和稠密视差图, 第 1 行的 4 幅图是各测试图对中的基准图像, 从左向右依次是 Tsukuba、Venus、Teddy 和 Cones。第 2 行的 4 幅图, 是各个基准图像在彩色分割后的分割结果 (白色表示分割区域边界, 位置邻近且灰度相同表示属于同一分割)。第 3 行的 4 幅视差图是各个图像对应的标准视差, 第 4 行是本文算法得出的视差, 而第 5 行是本算法所得视差与标准视差之间的差异 (误差大于误匹配阈值), 其中黑色的点是在无遮挡区域中计算错误的像素点, 灰色的点是在遮挡区域中匹配错误的像素点, 白色的大片区域均是匹配正确的像素点。图 3 有效且直观地反映了本算法的优劣。除了个别较大的低纹理区域 (分割结果内较大的个别区域), Mean-shift 分割获得的过分割, 总体上还是比较满意。从

图 3 可以看出,即使在这些大的低纹理区域,本算法也能够获得稠密且可靠的视差图;除个别高纹理、低纹理区域或互相交错的区域(如 Teddy 图中熊耳朵

旁)之外,绝大多数的立体匹配的难点区域如遮挡区域、低纹理区域和视差不连续区域,均能保持明显的视差边界并具有精确的视差值。

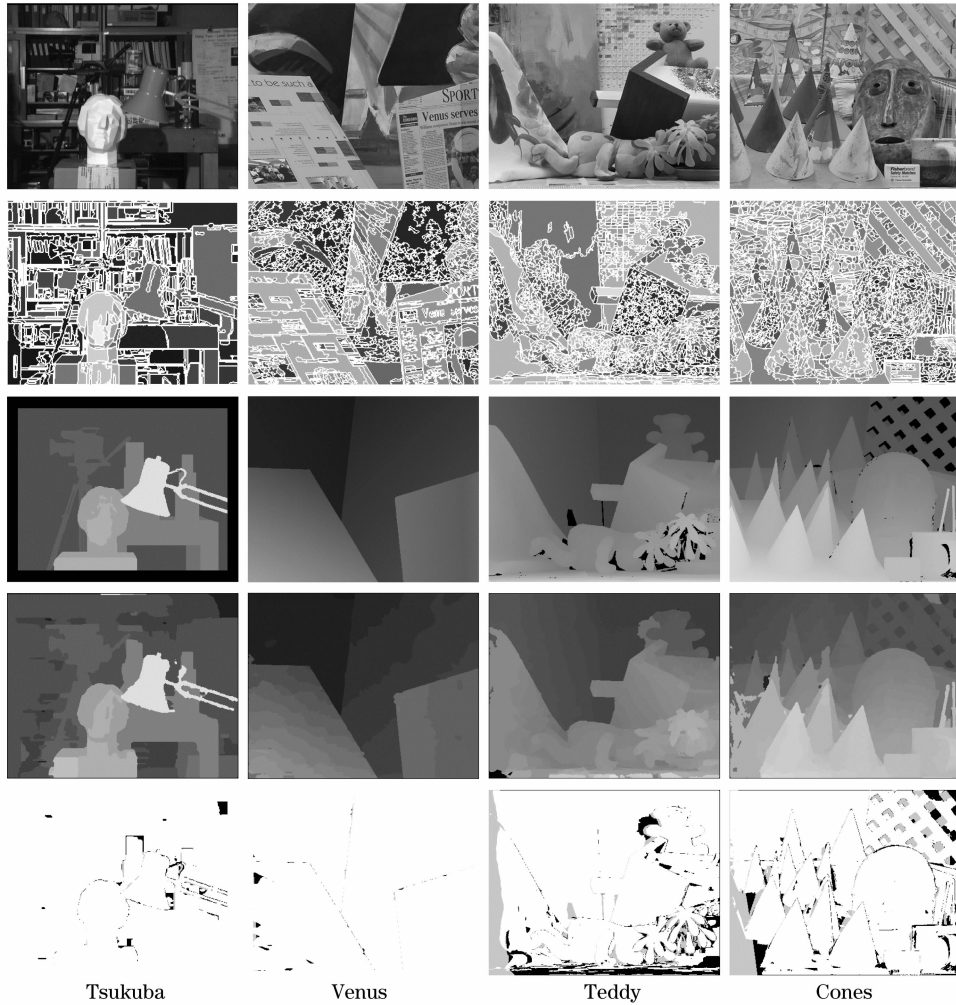


图 3 立体图像对‘Tsukuba’, ‘Venus’, ‘Teddy’ 和 ‘Cones’ 的分割结果和稠密视差。

从上到下各行依次是基准图像、分割结果、标准视差、本算法结果以及误匹配图像

Fig. 3 Segmentation information and dense disparity results for the ‘Tsukuba’, ‘Venus’, ‘Teddy’ and ‘Cones’ stereo image pairs. From top to bottom, each row represents base image,

segmentation result, ground truth, our result by the proposed algorithm and bad pixels, respectively

将本算法的结果传到 Middlebury 网站<sup>[2]</sup>进行了评估(2008 年 6 月 21 日)。表 1 为反馈回的基于新版的性能评估,显示的是误匹配率、算法排名等信息,其中  $R_o$  为在非遮挡区域的错误百分比,  $R_A$  是未知像素除外的总的错误百分比,  $R_D$  为在接近深度不连续区域的错误百分比。正粗体显示本算法结果。在部分列中,下标是相应列的相对排名;黑体表示该列中最小的错误百分比。在总共 45 个参与评测的算法中,本算法各项排名的平均值是 22.4,排在第 24 位,仅次于排在第 20 位的 AdaptWeight 算法<sup>[6]</sup>;目前仅列出其中 17 个结果

(版面有限)。算法 CooptRegion<sup>[13]</sup>、AdaptDispCalib<sup>[7]</sup>、CostAggr+occ<sup>[12]</sup>、SegmentSupport<sup>[9]</sup> 和 AdaptWeight<sup>[6]</sup> 是当前最好的前几个采用了局部图像信息的匹配算法。从正确匹配率来看,本算法稍差于上述几个利用了局部信息的算法。本算法的正确匹配率低于最新的基于全局的算法 AdaptingBP<sup>[3]</sup>、OverSegmBP<sup>[1]</sup> 及基于半全局的算法 C-SemiGlob<sup>[5]</sup>;但本算法不需要严格的图像先验知识、反复迭代,且速度较快。而本算法明显优于 GC+occ<sup>[14]</sup> 和 TreeDP<sup>[15]</sup> 等全局算法及 TensorVoting<sup>[16]</sup> 和 SSD+MF<sup>[2]</sup> 等局部算法。

表 1 基于新版的 Middlebury 立体评估结果  
Table 1 Results based on the new Middlebury stereo evaluation

Algorithm	Avg. Rank	Tsukuba			Venus			Teddy			Cones		
		$R_{\bar{\sigma}}$	$R_{\Lambda}$	$R_D$	$R_{\bar{\sigma}}$	$R_{\Lambda}$	$R_D$	$R_{\bar{\sigma}}$	$R_{\Lambda}$	$R_D$	$R_{\bar{\sigma}}$	$R_{\Lambda}$	$R_D$
CooptRegion	3. 0 <sub>1</sub>	<b>0. 87<sub>1</sub></b>	<b>1. 16<sub>1</sub></b>	<b>4. 61<sub>1</sub></b>	0. 11 <sub>2</sub>	0. 21 <sub>2</sub>	1. 54 <sub>3</sub>	5. 16 <sub>7</sub>	8. 31 <sub>4</sub>	13. 0 <sub>6</sub>	2. 79 <sub>3</sub>	<b>7. 18<sub>1</sub></b>	8. 01 <sub>5</sub>
AdaptingBP	3. 2 <sub>2</sub>	1. 11 <sub>7</sub>	1. 37 <sub>4</sub>	5. 79 <sub>8</sub>	<b>0. 10<sub>1</sub></b>	0. 21 <sub>3</sub>	<b>1. 44<sub>1</sub></b>	4. 22 <sub>4</sub>	7. 06 <sub>2</sub>	11. 8 <sub>4</sub>	<b>2. 48<sub>1</sub></b>	7. 92 <sub>3</sub>	<b>7. 32<sub>1</sub></b>
AdaptDispCalib	12. 8 <sub>9</sub>	1. 19 <sub>9</sub>	1. 42 <sub>5</sub>	6. 15 <sub>10</sub>	0. 23 <sub>10</sub>	0. 34 <sub>5</sub>	2. 50 <sub>12</sub>	7. 80 <sub>20</sub>	13. 6 <sub>23</sub>	17. 3 <sub>18</sub>	3. 62 <sub>13</sub>	9. 33 <sub>13</sub>	9. 72 <sub>16</sub>
C-SemiGlob	13. 4 <sub>11</sub>	2. 61 <sub>31</sub>	3. 29 <sub>26</sub>	9. 89 <sub>29</sub>	0. 25 <sub>13</sub>	0. 57 <sub>11</sub>	3. 24 <sub>17</sub>	5. 14 <sub>6</sub>	11. 8 <sub>9</sub>	13. 0 <sub>6</sub>	2. 77 <sub>2</sub>	8. 35 <sub>5</sub>	8. 20 <sub>6</sub>
CostAggr+occ	15. 6 <sub>14</sub>	1. 38 <sub>18</sub>	1. 96 <sub>18</sub>	7. 14 <sub>21</sub>	0. 44 <sub>17</sub>	1. 13 <sub>21</sub>	4. 87 <sub>21</sub>	6. 80 <sub>12</sub>	11. 9 <sub>11</sub>	17. 3 <sub>17</sub>	3. 60 <sub>11</sub>	8. 57 <sub>6</sub>	9. 36 <sub>14</sub>
OverSegmBP	15. 8 <sub>15</sub>	1. 69 <sub>24</sub>	1. 97 <sub>19</sub>	8. 47 <sub>26</sub>	0. 51 <sub>20</sub>	0. 68 <sub>16</sub>	4. 69 <sub>20</sub>	6. 74 <sub>11</sub>	11. 9 <sub>13</sub>	15. 8 <sub>9</sub>	3. 19 <sub>9</sub>	8. 81 <sub>10</sub>	8. 89 <sub>12</sub>
SegmentSupport	16. 2 <sub>16</sub>	1. 25 <sub>12</sub>	1. 62 <sub>8</sub>	6. 68 <sub>14</sub>	0. 25 <sub>12</sub>	0. 64 <sub>15</sub>	2. 59 <sub>13</sub>	8. 43 <sub>26</sub>	14. 2 <sub>24</sub>	18. 2 <sub>21</sub>	3. 77 <sub>15</sub>	9. 87 <sub>18</sub>	9. 77 <sub>17</sub>
AdaptWeight	19. 0 <sub>20</sub>	1. 38 <sub>18</sub>	1. 85 <sub>16</sub>	6. 90 <sub>18</sub>	0. 71 <sub>21</sub>	1. 19 <sub>22</sub>	6. 13 <sub>22</sub>	7. 88 <sub>21</sub>	13. 3 <sub>21</sub>	18. 6 <sub>26</sub>	3. 97 <sub>20</sub>	9. 79 <sub>16</sub>	8. 26 <sub>7</sub>
Our method	<b>22. 4<sub>24</sub></b>	<b>1. 78<sub>25</sub></b>	<b>2. 28<sub>23</sub></b>	<b>7. 00<sub>20</sub></b>	<b>0. 46<sub>18</sub></b>	<b>0. 74<sub>18</sub></b>	<b>2. 74<sub>15</sub></b>	<b>8. 06<sub>22</sub></b>	<b>13. 5<sub>22</sub></b>	<b>18. 4<sub>24</sub></b>	<b>5. 91<sub>31</sub></b>	<b>11. 9<sub>28</sub></b>	<b>11. 1<sub>23</sub></b>
RealtimeBP	24. 5 <sub>25</sub>	1. 49 <sub>21</sub>	3. 40 <sub>28</sub>	7. 87 <sub>24</sub>	0. 77 <sub>22</sub>	1. 90 <sub>28</sub>	9. 00 <sub>29</sub>	8. 72 <sub>28</sub>	13. 2 <sub>19</sub>	17. 2 <sub>16</sub>	4. 61 <sub>23</sub>	11. 6 <sub>26</sub>	12. 4 <sub>30</sub>
GC+occ	26. 0 <sub>27</sub>	1. 19 <sub>8</sub>	2. 01 <sub>21</sub>	6. 24 <sub>11</sub>	1. 64 <sub>31</sub>	2. 19 <sub>30</sub>	6. 75 <sub>23</sub>	11. 2 <sub>34</sub>	17. 4 <sub>33</sub>	19. 8 <sub>30</sub>	5. 36 <sub>29</sub>	12. 4 <sub>30</sub>	13. 0 <sub>32</sub>
Layered	26. 6 <sub>29</sub>	1. 57 <sub>22</sub>	1. 87 <sub>17</sub>	8. 28 <sub>25</sub>	1. 34 <sub>28</sub>	1. 85 <sub>26</sub>	6. 85 <sub>25</sub>	8. 64 <sub>27</sub>	14. 3 <sub>25</sub>	18. 5 <sub>25</sub>	6. 59 <sub>34</sub>	14. 7 <sub>32</sub>	14. 4 <sub>33</sub>
TensorVoting	29. 6 <sub>31</sub>	3. 79 <sub>35</sub>	4. 79 <sub>35</sub>	8. 86 <sub>27</sub>	1. 23 <sub>27</sub>	1. 88 <sub>27</sub>	11. 5 <sub>31</sub>	9. 76 <sub>30</sub>	17. 0 <sub>32</sub>	24. 0 <sub>36</sub>	4. 38 <sub>22</sub>	11. 4 <sub>25</sub>	12. 2 <sub>28</sub>
RealTimeGPU	29. 8 <sub>32</sub>	2. 05 <sub>28</sub>	4. 22 <sub>33</sub>	10. 6 <sub>32</sub>	1. 92 <sub>34</sub>	2. 98 <sub>32</sub>	20. 3 <sub>38</sub>	7. 23 <sub>17</sub>	14. 4 <sub>26</sub>	17. 6 <sub>19</sub>	6. 41 <sub>33</sub>	13. 7 <sub>31</sub>	16. 5 <sub>35</sub>
CostRelax	30. 9 <sub>33</sub>	4. 76 <sub>40</sub>	6. 08 <sub>38</sub>	20. 3 <sub>42</sub>	1. 41 <sub>30</sub>	2. 48 <sub>31</sub>	18. 5 <sub>37</sub>	8. 18 <sub>25</sub>	15. 9 <sub>30</sub>	23. 8 <sub>34</sub>	3. 91 <sub>18</sub>	10. 2 <sub>20</sub>	11. 8 <sub>26</sub>
TreeDP	33. 0 <sub>35</sub>	1. 99 <sub>27</sub>	2. 84 <sub>25</sub>	9. 96 <sub>30</sub>	1. 41 <sub>29</sub>	2. 10 <sub>29</sub>	7. 74 <sub>27</sub>	15. 9 <sub>39</sub>	23. 9 <sub>39</sub>	27. 1 <sub>40</sub>	10. 0 <sub>38</sub>	18. 3 <sub>37</sub>	18. 9 <sub>36</sub>
SSD+MF	40. 1 <sub>41</sub>	5. 23 <sub>43</sub>	7. 07 <sub>41</sub>	24. 1 <sub>43</sub>	3. 74 <sub>38</sub>	5. 16 <sub>38</sub>	11. 9 <sub>32</sub>	16. 5 <sub>41</sub>	24. 8 <sub>40</sub>	32. 9 <sub>43</sub>	10. 6 <sub>40</sub>	19. 8 <sub>40</sub>	26. 3 <sub>42</sub>

在初始匹配的匹配代价和视差估计的视差校准中,均充分地利用获得的分割信息(如过分割),加入色彩空间上的分割约束,从而比较适宜于景物表面邻近区域(如面积较小的面元)之间色彩对比比较明显的场景;同时,处理窄遮挡后,视差估计又基于多方向自适应加权最小二乘拟合填充宽遮挡等,所以也有利于处理存在较宽遮挡的场景。本算法上述优越性,从基于定性呈现的图 3 和定量分析的表 1 的综合评价,得到了有效地验证。

## 6 结 论

为了消除局部匹配的歧义性,提出一种基于 SASW 代价聚合的初始匹配和贪婪的视差估计后处理相结合的匹配方法。初始匹配,利用图像分割、对称地可变速权重策略,消除特征相似的匹配歧义性,构造了稳健的代价函数,从而得到较好的初始视差;后处理中贪婪的视差估计方案,有效地消除了低纹理、重复纹理和遮挡等复杂歧义,从而提高匹配精度。实验表明,基于 SASW 的初始匹配和贪婪后处理相结合的本算法,能获得分段平滑、有明显边界且精度高的稠密视差图,尤其对邻近景物面元间色彩区分度较大或有宽遮挡的场景。如何将本算法与其它匹配算法有机地相结合,进一步提高匹配性能,将

是今后研究的可能方向。

**致谢** 本文中所使用的 4 对标准测试图像和评估来自于 <http://vision.middlebury.edu/stereo>,在此对 Daniel Scharstein 和 Richard Szeliski 表示感谢。

## 参 考 文 献

- 1 C. Lawrence Zitnick, Sing Bing Kang. Stereo for image-based rendering using image over-segmentation [J]. *International Journal of Computer Vision*, 2007, **75**(1): 49~65
- 2 Daniel Scharstein, Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms [J]. *International Journal of Computer Vision*, 2002, **47**(1/ 2/ 3): 7~42
- 3 Andreas Klaus, Mario Sormann, Konrad Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure [C]. *IEEE International Conference on Pattern Recognition*, 2006. 15~18
- 4 Stan Birchfield, Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998, **20**(4): 401~406
- 5 Heiko Hirschmüller. Stereo processing by semiglobal matching and mutual information [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008, **30**(2): 328~341
- 6 Kuk-Jin Yoon, In So Kweon. Adaptive support-weight approach for correspondence search [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006, **28**(4): 650~656
- 7 Gu Zheng, Su Xianyu, Liu Yuankun et al.. Local stereo matching with adaptive support-weight, rank transform and disparity calibration [J]. *Pattern Recognition Letters*, 2008, **29**(9): 1230~1235

- 8 Gu Zheng, Su Xianyu. An algorithm based on adaptive support weight and disparity adjustment for trinocular stereo matching [J]. *Acta Optica Sinica*, 2008, **28**(4): 734~738  
顾征, 苏显渝. 三目自适应权值立体匹配和视差校准算法[J]. *光学学报*, 2008, **28**(4): 734~738
- 9 Federico Tombari, Stefano Mattoccia, Luigi Di Stefano. Segmentation-based adaptive support for accurate stereo correspondence[C]. *IEEE Pacific-Rim Symposium on Image and Video Technology*, 2007. 427~438
- 10 Jong Dae Oh, Siwei Ma, C.-C. Jay Kuo. Stereo matching via disparity estimation and surface modeling [C]. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2007. 1696~1703
- 11 Dorin Comaniciu, Peter Meer. Mean shift: A robust approach toward feature space analysis[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, **24**(5): 603~619
- 12 Dongbo Min, Kwanghoon Sohn. Cost aggregation and occlusion handling with WLS in stereo matching[J]. *IEEE Transactions on Image Processing*, 2008, **17**(8): 1431~1442
- 13 Wang Zengfu, Zheng Zhigang. A region based stereo matching algorithm using cooperative optimization [C]. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2008. 887~894
- 14 Vladimir Kolmogorov, Ramin Zabih. Computing visual correspondence with occlusions using graph cuts [C]. *IEEE International Conference on Computer Vision*, 2001. 508~515
- 15 Olga Veksler. Stereo correspondence by dynamic programming on a tree[C]. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2005. 384~390
- 16 Philippos Mordohai, Gérard Medioni. Stereo using monocular cues within the tensor voting framework[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006, **28**(6): 968~982

## 《激光与光电子学进展》“光学制造”栏目征稿启事

《激光与光电子学进展》是中国科学院上海光学精密机械研究所主办的激光、光电子领域行业性期刊,中国科技核心期刊。该刊1964年创刊,至今已出刊500余期,旨在促进国内外学术交流,沟通科研单位、生产部门与用户的联系。2009年,《激光与光电子学进展》将加大光学制造方向的报道力度,深入研讨光学制造的技术发展及其在工业、科研各方面的应用。主要涉及方向有光学元器件、光学仪器、光学加工、光学设计、光学材料、光学薄膜、光学检测等领域。

欢迎相关的科研、技术、市场人员根据我们栏目内容定位,撰写或向我们推荐涉及上述领域的优秀技术论文。来稿不收取审稿费和版面费,一经录用将优先发表并支付丰厚稿酬。

• 本栏目基本要求:

- 1) 稿件内容应为光学制造领域的最新进展、研究动态、科研和新品开发成果等方面;
- 2) 文章要求内容新颖、论点正确、论据充分、数据可靠、文理通顺;
- 3) 技术文章字数一般为5000字左右;综述文章字数最多不超过10000字,且第一作者需为副教授及以上职称。

• 投稿方式:通过网上投稿系统(<http://www.opticsjournal.net/lop.htm>)直接上传稿件(主题标明“光学制造”投稿),也可以直接将稿件word版发至邮箱:[dingjie@siom.ac.cn](mailto:dingjie@siom.ac.cn)(主题标明“光学制造”投稿),如有问题欢迎垂询021-69918198与栏目编辑丁洁联系。

《激光与光电子学进展》编辑部