## 基于匹配代价卷融合和多步插值策略的立体匹配

## 1.摘要

针对双目立体匹配算法中的自适应窗口法在重复纹理区表现不好的问题提出了一种基于匹配代价卷融合的策略；在后处理步骤中，提出一种基于分割的插值策略，提高了图片左右边界处的匹配精度，并且提出一种联合三种插值策略的方法，比已有的双插值策略匹配精度更高。与其他经典立体匹配算法的对比证明了本方法在计算视差精度上的优势。

## 引言

立体匹配通过搜寻左右图片对的对应像素点来计算视差，根据三角相似性原理，从视差值可求出物体到相机的距离。立体匹配算法被广泛应用于自动驾驶，智能机器人，Slam等领域中，是计算机视觉领域的研究热点[参考文献]。根据\*\*的分类，立体匹配算法被分为全局立体匹配和局部立体匹配[ ]。全局算法将视差计算表述为一个能量最小化的问题，可以通过置信度传播、图割等算法来求解，匹配精度高，但计算耗时；局部匹配算法通过为目标像素点构建局部支持窗口，利用邻域信息来降低匹配的模糊性，具有实现简单，运行速度块的优点，一些局部算法的精度可以媲美全局算法，使得局部匹配算法广泛地应用于现实任务，尤其是实时任务中。局部立体匹配流程通常被分为四个步骤：匹配代价计算、代价聚合、视差计算（也称为视差优化或视差细化）。代价聚合步骤是提高匹配精度的关键，受到了广泛的研究。

代价聚合利用局部支持区域的信息来降低匹配的歧义性，也可以把它看作对匹配代价卷的每一个视差层的滤波操作。局部支持区域构建的好坏直接决定聚合的效果，支持区域用窗口来表示，其假设窗口内的视差是平滑性的，当窗口覆盖两个或者多个视差平面时，聚合操作会引起前景区域的扩展或者缩小【多窗口法2002】，为保证视差平滑性必须限制窗口过大，另一方面，窗口太小会使得局部支持信息不足也达不到理想的匹配效果。针对这一矛盾，许多学者对于窗口的构建策略进行了研究。

【多窗口法】【移动窗口法】等。相较固定窗口，这两个方法大大提高了在视差不连续区的表现，但是由于窗口形状种类有限制，对物体边缘的贴合程度不够，结果仍不能让人满意。

Zhang等人[ ]提出了基于十字交叉的自适应支持窗口构建策略，通过颜色和距离限制构建目标像素点的水平垂直臂进而构建整个支持窗口，对视差边缘有较好的检测能力。Mei等人对此进行了改进，根据臂长的长度对颜色限制强度对进行分段，在不引入误差的同时增大了支持窗口面积 [ ]。但是，该类方法有一个缺陷：重复纹理区频繁的颜色变化会使在该区域构建的窗口过小，从而增加匹配的歧义性。

视差细化运用一系列图像处理方法来纠正初始视差图中的错误值，其中，插值是使用最频繁的策略，对于遮挡区域的视差估计必须通过插值方法来求得。背景插值简单高效，鲁棒性低于区域投票策略高于合理插值策略，区域投票对支持区域的合理值和最高投票比率有最低值限制，导致其对于遮挡区域的插值效率不高，合理插值在背景插值的基础上增大了搜素方向和深度，插值效率高但是增大了误差。

基于十字交叉的代价聚合策略对视差不连续区取得了较好的效果但是在重复纹理区域误匹配率较高；插值策略很多，但各有局限性；另外，针对图像左右边缘处遮挡区的视差估计也缺少特定的插值策略，匹配精度有待提高。

针对这些问题，本文提出一种融合十字交叉自适应窗口和固定窗口的代价聚合策略，比单独使用自适应十字交叉窗口在重复纹理区表现好，并且提出一种联合背景插值和区域投票插值的策略，比单独使用区域投票得到了更低的误匹配率。此外，为提高在图像边缘的匹配效果，提出一种基于分割的插值策略。

## 2.本文算法

本文提出的立体匹配方法流程如图1所示。代价函数以性能较优的AD-Census为基础，主要在代价聚合、视差细化方面提出更有效的算法。

Local stereo matching with improved matching cost and disparity refinement

2.1视差计算

WTA(Winner-Takes-All)策略通常被用来进行视差计算，其公式如下：

(11)

式中，为目标像素，为视差，为代价值，为视差，为设定的最大视差。

该策略操作简单，执行速度快，能够求得最小代价对应的视差。但是，在弱纹理和重复纹理区，会出现正确视差的代价不是最小且存在多个接近的小代价值的现象；另外，噪声也会使最小代价值和正确视差不对应。WTA策略不能正确处理上述两种情况。

针对上述问题，我们利用邻域信息来降低视差选择的歧义性。首先，我们从匹配代价卷构建每个点的候选视差集，若候选视差集中的元素差距过大，说明该点的视差存在较大不确定性，需要借助邻域已经确定的视差来辅助该点的视差选择，即从候选视差中选择和已确定的邻域视差最接近的视差；假定多数情况下邻近区域的视差是相近的，当候选视差集中存在视差接近的元素时，我们认为正确视差存在于这些视差中，而距离这些接近的视差较远的单个元素，则判定是误差，将其剔除，当邻域确定视差和目标点的候选视差集接近时，引入邻域的视差信息来辅助选择，统计目标点和邻域点的候选视差集合中目标点候选视差的数量，数量最多的视差为最终视差。具体算法流程见算法1

算法1

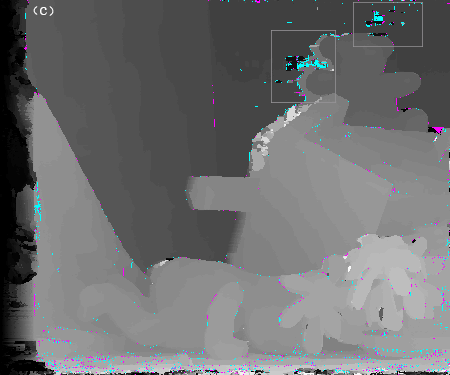
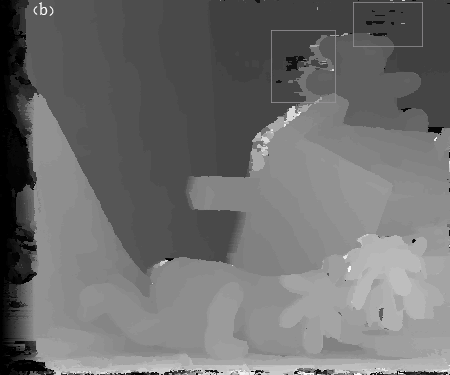
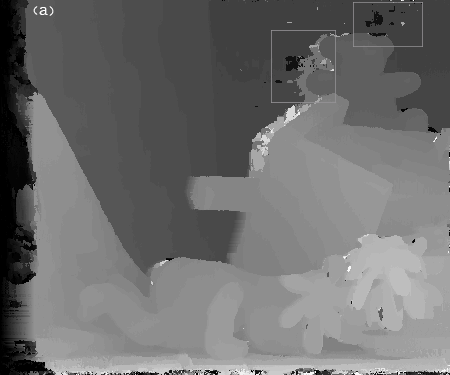
1. 记录匹配代价卷每个图像坐标下的最小代价值的N%范围内的前M个视差，记为候选代价卷V，V(u, v)表示坐标(u, v)的候选视差集，其中u表示行号，v表示列号，Vi(u, v)表示V(u, v)中的元素，Ci(u, v)表示Vi(u, v)的代价，N(u, v)表示V(u, v)中元素个数,D(u, v)表示(u, v)点最终确定的视差。
2. 若u==0或者N(u, v) ==1, 则D(u, v) = , 否则进行(3);
3. 设 若S < T1, 则将和存入集合H中，相同元素只存一次，用Hi表示H中的元素, Pi表示Hi的代价
4. 统计H中元素的个数n，若n==0，则, 否则，进行(5)
5. 若不存在 , 清空H，否则，进行（6）

统计H, V(u-1, v), V(u + 1, v)中Hi的数目Ni和对应的代价和Xi，, 若h唯一，则D(u, v) = h, 否则，清空H

设置N为105，M为4，图1（a），（b）分别是在本文框架下执行WTA策略和算法1获取

图1 不同视差计算策略下的视差图。（a）WTA的视差图；（b）算法1的视差图；（c）WTA和算法1的视差对比图；（d）原始图像

Fig.1 Disparity maps of different disparity calculating stratagies. (a)Disparity map of WTA; (b)Disparity map of algorithm 1;(c)Disparity comparision map of WTA and algorithm 1;(d)Original map

的视差图，（c）是（b）和（a）的视差对比图，其中蓝色区表示(a)图视差错误而（b）图视差正确的区域，红色区表示（a）图视差正确而（b）图视差错误的区域，其他区域表示（b）和（a）的正确性一致。可以发现，在重复纹理区（框选区域）算法1比WTA有更少的误匹配，表明了算法1的有效性。

2.2视差后处理

经过上述步骤后，在遮挡区和视差不连续区会存在大量的误匹配点，将它们检测出后需要利用邻域的视差进行插值填充。首先，利用左右一致性检测标记出左视差图的误匹配点，并进一步分为遮挡点和误匹配点：

(12)

式中，为左视差图，为右视差图，当像素p满足（12），则将其标记为合理点，否则，记其为误匹配点并做下一步划分，若d,满足公式（13）：

(13)

则将点记为误匹配点，否则记为遮挡点。

对于误匹配点，采取两轮不同的插值策略：第一轮，进行区域投票插值【CBCA】插值，区域投票利用自适应窗口中的频率最高的合理视差来进行插值，虽然可靠性高，但是由于位于遮挡区的点的自适应窗口较小并且含有的合理视差数目较少，导致区域投票策略可靠性下降，插值效率低，所以对于遮挡点，我们采取背景插值策略：

(14)

式中，，分别表示p点左侧和右侧第一个合理点，用限制搜寻的范围来控制可靠性。

对于误匹配点，采取区域投票插值策略，直接用前面构建的自适应窗口，所以只需要统计该点所在窗口中的视差直方图

## 4.实验结果与分析

实验平台配置为Intel(R) Core(TM) i5-7300U CPU，2.60GHz，8GB内存。利用visual studio 2019IDEA，使用C++语言编程实现算法。数据集选用Middleburry基准测试平台的第二版的数据集上，包括Teddy、Cones、Venus、Tsukuba四对图片，并设置视差搜索范围分别为0~59、0~59、0~19、0~15px。

评价标准和Middleburry平台保持一致，即误匹配像素百分比，并且设置误差限为1个像素，当某个像素点的预测视差值和真实视差值的差距大于1时，该点被识别为误匹配像素，分别计算在非遮挡区、所有区域和视差不连续区的误匹配像素百分比，区域的选择通过掩膜图来实现，掩膜图为二值灰度图，分别为0和255，其中255表示需要计算的点，而0表示不计算的点。

4.1 匹配代价卷融合策略的有效性

为了更好地观察匹配代价卷融合策略的效果，实验只保留匹配代价计算、代价聚合步骤，通过WTA获取视差图，不进行后处理步骤。设置参数，总臂长长度，单臂长长度。在代价聚合步骤中，分别使用基于十字交叉窗口的代价聚合和本文提出的代价卷融合策略，结果如表6 所示。

不同代价聚合方法的误匹配百分比

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Aggregation method | Tsukuba | | | Venus | | | Teddy | | | Cones | | | Ave |
| no-occ | all | disc | no-occ | all | disc | no-occ | all | disc | no-occ | all | disc |
| CBCA |  |  |  | 0.62 | 2.13 | 5.10 | 6.26 | 15.83 | 16.08 | 4.59 | 15.20 | 11.76 |  |
| Proposed |  |  |  | 0.58 | 2.05 | 4.63 | 6.25 | 15.79 | 15.99 | 4.53 | 15.06 | 11.61 |  |

结果如下，

两个方法的精度表

结果图

时间对比

对结果进行讨论

4.2 针对特定区域插值策略的有效性

4.2.1 背景插值和基于区域投票的联合插值策略的有效性

对比联合插值策略和区域投票的结果

4.2.2 基于图像分割的图片左边缘插值的有效性

在上一个步骤后，跟这个步骤，对比两者的结果

4.3 总的结果的对比

对比该方法和其他经典方法的结果，数据的和图片的

## 5.结论