## 基于新的代价计算方法和联合插值策略的立体匹配优化

## 1.摘要

在这篇论文里，我们提出一种局部立体匹配方法，它采用了新的视差计算策略和一种联合插值视差优化策略。先利用census和梯度相结合的方法计算初始匹配代价。用基于十字交叉的自适应窗口法进行代价聚合。采用一种结合邻域信息的视差计算策略来计算初始视差。最后进行视差后处理。利用Middleburry平台的图片集进行多组对比实验，证明了本文算法的有效性。（这边可以具体点，比如在哪些位置）

## 引言

立体匹配是计算机视觉领域的研究热点，在许多应用场景，如虚拟现实，机器人，自动驾驶中发挥着重要作用【a】。它通过匹配立体图像对的像素点来获取视差图。根据【b】，立体匹配算法可分为全局匹配和局部匹配。全局算法将视差计算转化为能量最小化问题，可以通过置信度传播【c】、图割【d】等算法来求解，匹配精度高，但计算耗时；局部算法利用局部支持窗口来降低匹配的模糊性，实现简单，运行速度快。局部立体匹配可分为代价计算、代价聚合、视差计算、视差后处理四个步骤。

代价计算用于计算初始匹配代价，方法包括：AD、BT、Census、NCC、梯度、互相关等。近些年代价计算上的创新集中于对已有单个方法的融合，如AD-census【ADcensus】、AD-gradient【guideFilter】、Census-MI-gradient【e】，AD-Census-Grad，它们融合多个维度的信息来提高代价计算的准确率。代价聚合通过计算区域内初始匹配代价的加权均值来降低匹配的歧义性，其面临一个折衷问题：如何在尽可能少得包含视差不同于目标点的点的同时包含更多的视差同于目标点的点，有两个解决方案，一是采用自适应加权平均法，【ASW】【GF】，根据支持点和目标点的相关程度来计算支持区域内每个点的权重；二是采取自适应窗口法，通过一定规则改变支持区域的形态，如基于十字交叉的自适应窗口法【Cross-Base】，和【ASW】相同的思路，它也把颜色和距离作为了邻近像素和目标像素相关程度的度量，并以此构建和物体边缘贴合的窗口。

WTA(Winner Takers All)策略选择最小代价值对应的视差作为视差计算的结果，由于其简单性，得到广泛的使用【】【】，然而在弱纹理和重复纹理区的点会存在多个接近的最小代价值的情况，使得WTA不可靠。郭鑫（2019）等提出的可靠性差别视差修正算法，先得出最小和次小匹配代价，并判定这两个代价的差和商是否大于给定阈值，若小于阈值证明视差值不可靠，对于不可靠的像素点，以该点为中心建立5x5矩形窗口，搜索窗口内的距离和颜色最接近的点，并把找到点的视差作为更新后的视差。这种方法的问题很明显，首先不同图像的阈值可能不同，其次在搜索窗内找到的点也不见得是可靠点。

本文关注立体匹配中代价计算和视差计算方法。首先，针对census和梯度的代价计算分别提出改进算法，并融合二者得到初始代价值，其效果好于ad-grad法，和ad-Census法相当；第二，提出了一种结合邻域的视差计算策略，对比WTA策略，使重复纹理区的误匹配明显降低。

## 2.代价计算与代价聚合方法

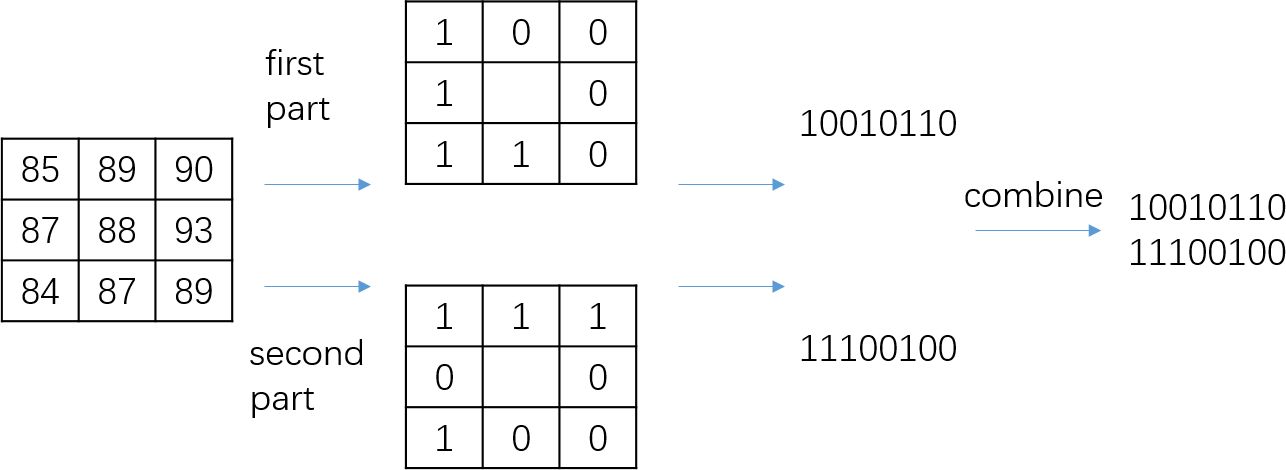
由于本文针对局部立体匹配中的视差计算和视差优化方法进行改进，因此需要首先对广泛使用的代价计算和代价聚合方法进行介绍。并且，这些方法将在实验部分用于分析和验证本文所提出的后处理方法的有效性。

### 2.1 代价计算

这个步骤通过计算像素描述符的差异或者相似性来获取初始匹配代价卷。【Evaluation of stereo matching costs on images with radiometric differences】指出，census在局部和全局方法中都显示出了最好的总体结果。Census通过比较目标点和周围点的强度值大小来构建目标点的编码，并以两个点编码的汉明距离作为匹配代价：



式中，为左图中一点， 表示点的邻域，表示把二进制位连接起来，表示点的census编码，表示点在视差为时的匹配代价，表示点视差为时在右图中的对应点，设点坐标为，则q点坐标为。

编码能反应图像的局部结构，这让census对光照变化鲁棒性高，但其过于依赖中心点存在隐患，当中心点被干扰时，编码会发生大范围变化，另一方面，只比较中心点仅能反应局部结构的一个方面，即中心和周围的关系。我们提出一种改进的census编码方式，编码由两部分组成，第一部分用传统编码方式生成，即通过比较中心点和周围点的相对大小，第二部分通过对紧挨中心点的8个邻域点按照顺时针顺序依次比较前后两个点的强度值大小获取，左上角点为起始点，过程如下：

相比传统方法，该方法增加了对不依赖中心点值的局部结构的描述，能降低中心点值被干扰时造成的误差，同时能更有效地利用局部信息。

梯度是相邻像素点强度变化的表征，包含了图像的纹理信息，对于点，我们用下式计算梯度：



和分别表示点横向和纵向的梯度，表示灰度值。基于梯度的代价计算对光照变化和噪声都有较好的鲁棒性【祝的改进census】，【基于颜色和边缘信息】【基于跨尺度引导滤波图像的稠密立体匹配】【Nolocal】【跨尺度】都使用了【GF】中的代价计算方法，即AD代价和梯度代价的结合，但其梯度代价只利用了横向梯度信息。从理论上分析，包含横向和纵向梯度信息的代价计算会比单独一个方向的表现更好，【祝的三代价结合】中的结合方式如下：



【基于改进代价计算和自适应引导滤波的立体匹配】也用了相同的结合方式，即对横向和纵向的梯度代价取相同的比重，这种做法没有考虑两个方向梯度的相对关系。本文提出一种利用十字支持臂长【ADCensus】对横向和纵向的梯度代价值的自适应加权结合策略，对可靠性高的方向梯度代价赋更高的权重。首先计算图像中每个点的十字支持臂长，计算可以看作从中心点出发的臂的生长过程，以左臂为例，用表示臂上最前端的点，表示点右边相邻的一个点，表示臂长，当不满足下式时，臂长停止增长：



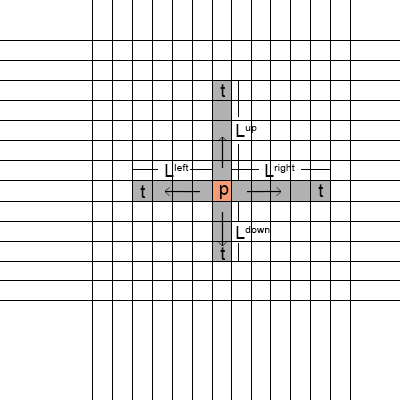
其中，为最大臂长，限制臂上点和中心点的距离在一定范围内，和是中心点和臂上点颜色差的限制阈值，的值小于，目的是对离中心点更远的点施加更加严格的颜色限制，这能在增大臂长范围的同时降低引入不同视差点的可能。第四个式子限制臂上相邻两点的颜色差，进一步增加了臂上点和中心点属于相同视差层的可靠性。另外三个方向的臂长计算和此类似，规定最短臂长为1，图示？。

我们利用水平方向和竖直方向最短臂长的比例关系作为梯度代价结合时的权重：

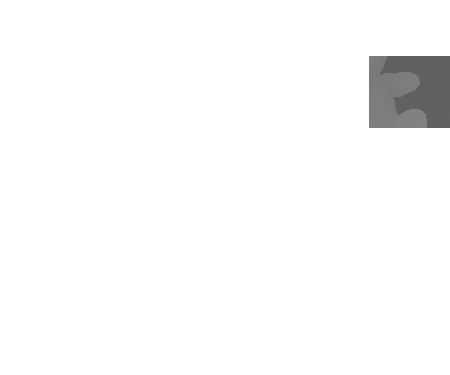
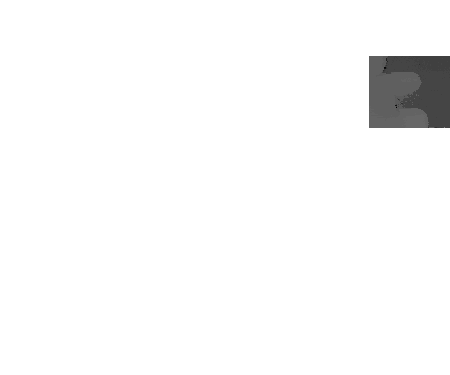
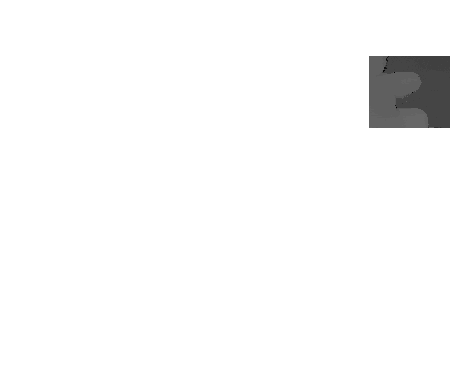


梯度代价结合公式为：





某方向的臂长越长，说明该方向上的点和中心点处于相同物体的可能性越大，为该方向的梯度代价赋更大的权重能提高物体边缘区视差估计的可靠性，如图所示【teddy在熊猫右侧的图】，非加权方法在边界出现更多的误匹配，可以看出自适应加权策略在梯度代价结合中的有效性。



将改进的census代价和改进的grad代价进行融合，得到初始匹配代价：



式中为控制参数，该融合方式借鉴了AD-Census，通过指数将不同代价归一到范围内，统一量纲，降低极端值的影响，并且通过调节两个代价对总代价的影响。

### 2.1 代价聚合

由于相似区域，噪音等因素的存在，单个点的匹配代价可靠性低，需要计算其所在支持窗口内所有点的初始匹配代价值的统计学量，如均值作为该点新的代价值：



式中，表示点的支持窗口，表示和间权重，用来控制点代价值对结果的贡献，若求均值，则取1，除以权重总和是为了规范化结果。

本文采用【ADCensus】中的改进版的基于十字交叉的自适应窗口的代价聚合算法，因为在上节已经描述了十字支持臂长的计算过程，所以这里直接对臂长进行扩展来构建支持窗口，简单地，只需计算垂直臂上每个点的左臂和右臂的长度，或者计算水平臂上每个点的上臂和下臂的长度，就能确定支持窗口，过程如图？所示，如果用分别表示左、右、上、下臂的长度，和分别表示水平臂和竖直臂上的点的集合，则点的支持窗口可通过如下公式表示：



其中，



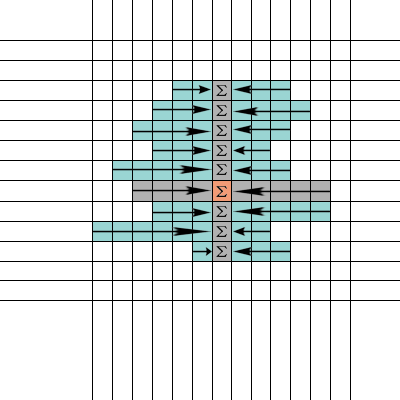
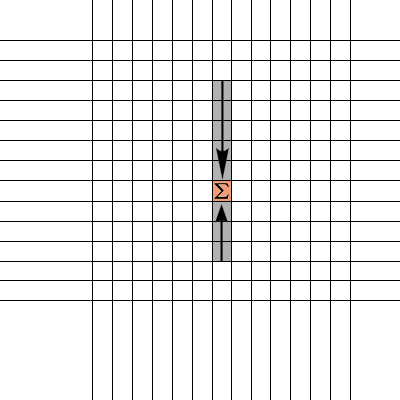
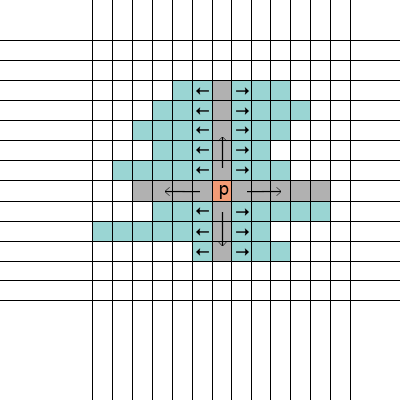
为保证支持窗口在左右图都满足颜色距离限制条件，取两者的交集作为最终的支持窗口：



在求取目标点的聚合代价时，采用一维积分图技术来加速计算过程，积分方式有两种，先水平积分再竖直积分，如图？，或者先竖直积分再水平积分，两者的聚合区域并不相同，【ADCensus】交替使用两种积分方式进行了4次代价聚合的迭代，我们在实验中发现，后三次迭代对视差错误率降低较小，且考虑到该方法迭代一次的高耗时，所以本文只使用一次先水平积分再竖直积分的迭代。

该方法实际是计算支持窗口内初始代价的均值作为新的代价值，可表示为:



式中，表示支持窗口包含的点的数目。 

### 2.3结合邻域信息与候选视差的视差计算

视差计算用来生成初始视差图，常用的WTA策略的做法是找到最小代价对应的视差：



式中，表示设定的最大视差，该策略操作简单，执行速度快，但是易受噪声影响，尤其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。针对上述问题，我们利用邻域信息与候选视差来降低视差选择的歧义性。

本文首先通过统计分析验证“候选视差”这一思路的可行性。在双目匹配图像中截取视差错误较大的部分，图？中两条横线中间的部分；先找到每个点的最小代价，然后选出范围内最小的4个匹配代价值，并取得其对应的视差；最后，统计这些候选视差与真实视差的匹配率。结果如表？所示，其中的代价不大于的代价。可以看出，匹配代价最小的视差（候选0）正确率53.3%，即WTA的结果，候选1-3的正确率总和为21.9%。根据统计值，如果能够利用候选视差来修正，理论上可以将视差的正确率提升到75.2%。但需要解决的问题是：如何在最小代价视差和其它候选视差中做出最优选择？



表3-1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Candi.0 | Candi.1 | Candi.2 | Candi.3 | Other | Sum |
| Num | 2136 | 485 | 239 | 151 | 1000 | 4011 |
| proportion（%） | 53.3 | 12.1 | 6 | 3.8 | 24.9 | 100 |

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。基本思想：考虑视差从左往右、从上往下应该是连续变化的，对于当前像素有多个候选视差难以确定的情况，可以借助邻域点的已确定视差或候选视差来辅助判别当前像素的最优视差。下面以点为例描述算法。

定义符号：设表示候选视差集合，表示候选视差，表示视差间的差距，表示包含元素的个数，表示该策略得出的视差。

算法描述：首先，从的所有视差中选出代价值最小的前M个视差作为候选视差，候选视差满足以下两个条件：

接下来，考虑候选视差选择的两种情况，分别定义规则：

情况1：的候选视差数为1，直接采用最小代价对应的视差：



情况2：的候选集元素数大于1，引入相邻8个点的候选视差来辅助点的判断，对中的每个元素，统计其在中和8个相邻点的候选视差中出现的总次数，并计算出现的视差的代价和，若存在唯一的最大总次数，则



否则，从选出的多个最大总次数对应的视差中，选择代价和最小的。

**3．2视差后处理**

初始视差图存在很多错误点，大多数位于遮挡区，需要检测出它们并用插值策略来修正。我们主要采用【AD-Census】中的视差后处理方法，依次进行左右一致性检测，基于区域投票的插值，16方向合理插值和中值滤波。首先，利用左右一致性检测找出错误点，若点不满足，被判为错误点，否则，被判为合理点，进一步对错误点进行分类，若点的对极线上存在和点匹配的点，则将点判为误匹配点，否则判为遮挡点。然后，对错误点所在十字型支持窗口内的合理点视差进行直方图投票，投票总数记作，得票数最高的视差记作，若满足以下两个条件， ，错误点的值更新为。

接着，对剩余的错误点执行16方向合理插值，即从错误点周围16个方向寻找每个方向上的第一个合理点，对于遮挡点，从中选择最小视差更新其值，因为背景区的视差小于其前景区，而遮挡点位于背景区，对于误匹配点，从中选择和其在原图上的颜色差最接近的点的视差。最后，对视差图执行的中值滤波以平滑噪声。

## 4.实验结果与分析

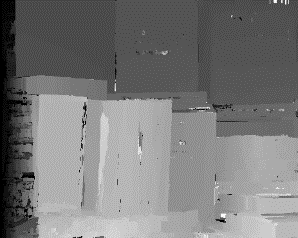
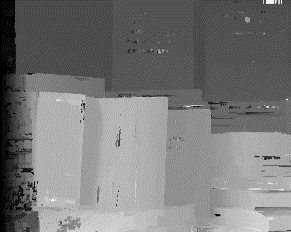
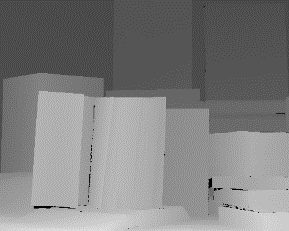
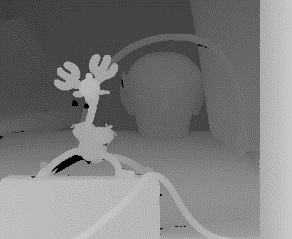
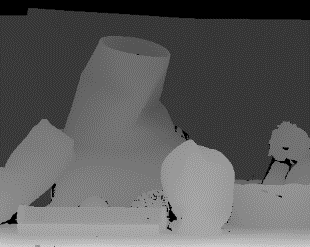
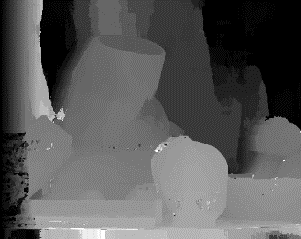
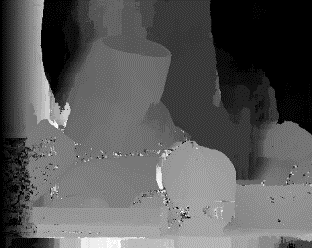
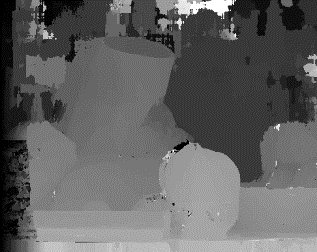
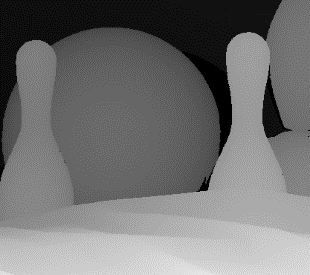
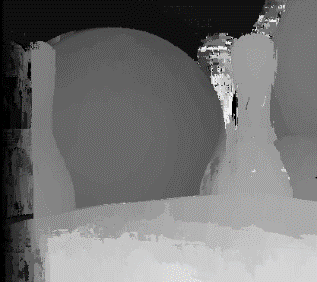
我们使用Middlebury立体数据集进行实验，为了能够在我们的设备上运行，所有图片均使用其提供的最小尺寸。错误阈值定为一个像素，用表示真实视差，表示计算得到的视差，若，则被判为错误值，在规定区域内，错误值的总数除以总的有效点数（排除不存在真实视差值的点），则得到误匹配率EMR，将该指标作为评价算法性能的指标。参数设置如下：

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter |  |  |  |  |  |  |  |  |  |
| Value | 17 | 34 | 20 | 6 | 20 | 15 | 0.4 |  |  |

**4.1 代价计算的有效性**

为验证提出的代价计算方法的有效性，分别选用AD-Census、AD-Grad和本文算法进行代价计算，采用上面描述的代价聚合算法进行代价聚合，利用WTA计算视差，表x为结果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Books | Reindeer | Midd1 | Rocks1 | Wood2 | Baby2 | Cloth2 | Bowling1 |
| AD-Census | 25.2 | 31.5 | 42.7 | 14.1 | 16.7 | 17.6 | 17.6 | 31.1 |
| AD-Grad | 27.5 | 33.4 | 44.4 | 14 | 16.9 | 22.2 | 20.4 | 35.1 |
| Proposed | 22.5 | 30.2 | 31.5 | 12.8 | 15.7 | 16.1 | 17.7 | 25.7 |

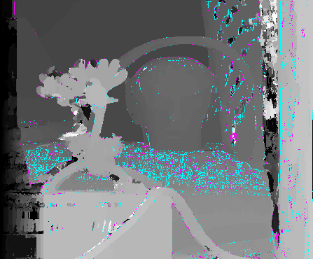
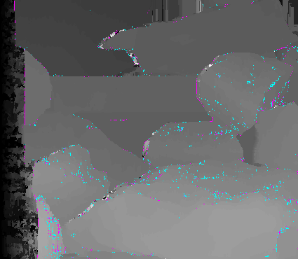
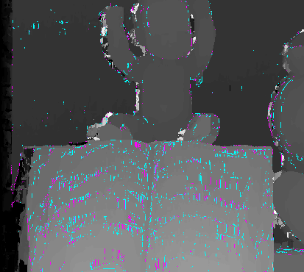
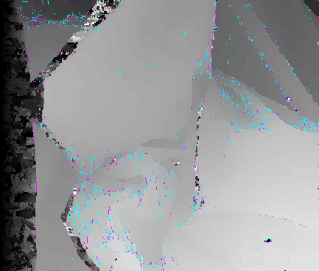
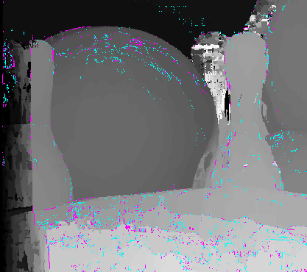
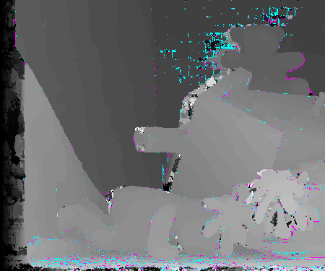
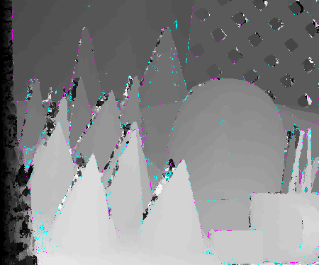
分析：从表？能够发现，除了在Wood1和Cloth2中，所提算法错误率略微高于AD-Census外，其他均好与AD-Census和AD-Grad。观察图？，可以看到，所提算法在弱纹理区比AD-Census和AD-Grad表现更好，分析有两方面原因，一是在census中增加了额外的内圈编码，使得census对局部结构的表征更加精确，二是AD本身在弱纹理区表现不好，而Census和梯度都是对局部结构的描述，能够好地处理弱纹理区域。

**4.2 视差计算策略的有效性**

为了验证所提出的结合邻域信息的视差计算策略的有效性，选择WTA算法进行对比实验，结果如表？所示。从结果能得出，所提算法生成的视差图精度高于WTA。从图中可以看到，所提算法在重复纹理和弱纹理区域能够获取非最小代价视差的正确视差，同时发现，该算法也存在误判，即为本来正确视差是最小代价视差的点选择了非最小代价视差，误判很多发生在背景为弱纹理的物体边界的背景区一侧，因为弱纹理区点的代价可能在很大的视差范围内都很接近，位于背景区的边界处的点会因为引入前景视差值而使得结果更向前景物体的视差值倾斜。在物体内部也有误判发生，但误判数目小于改正数目。

|  |
| --- |
|  |

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithm | Books | Reindeer | Midd1 | Rocks1 | Wood2 | Baby2 | Cloth2 | Bowling1 | Teddy | Cones | AVE |
| WTA | 22.5 | 30.2 | 31.5 | 12.8 | 15.7 | 16.1 | 17.7 | 25.7 | 16.2 | 15.4 | 21.5 |
| Proposed | 21.9 | 28.4 | 31 | 12.4 | 14.4 | 15.1 | 17.1 | 25.2 | 15.3 | 14.2 | 20.7 |

****

**4.3 参数分析**

**4.3 立体匹配算法总体性能对比**

**为进一步验证本文所提算法的表现，选择？？？算法进行对比，可以法向**

## 5.结论

问题1：结合邻域的

我看到有论文里是对比不同误差精度下的性能



郭鑫 2019 三种测度融合的加权引导滤波匹配算法＊