# 动态视差范围半全局双目立体匹配
# 期末大作业报告

电子学院 171830578

张凌宇

指导老师：李杨

2020年6月

# 摘要

基于双目视觉的立体匹配在无人驾驶、虚拟现实等诸多三维重建等场景中有着重要的应用。通过两个平行位置拍摄的图像，找到同名点，利用几何关系可以计算出深度信息。

本作业主要重现了DDR-SGM工作[1]，引入了一种新的代价函数的讨论，以及一种新的评价方式的设计，总结了当前基于深度学习的立体匹配方法。基于双目视觉的立体匹配传统的方法是半全局的能量最小化，使用Census代价计算匹配度，通过多路径聚合来平滑[2]，达到近似全局最优的视差图。DDR-SGM针对视频序列的立体匹配问题引入对时间上的相关性的利用。以前帧视差图为基础，在其相邻范围内进行下一帧视差图的视差搜索，在精度损失甚微的情况下大大提高了计算效率。

本文也讨论了一种将像素值分级的代价函数，相比Census将保留更多的邻域信息。这种代价函数计算效率低于Census，但一定程度上提高了匹配精度，在bad2.0和bad4.0上都有更好的表现。

本文还讨论了一种关注细节的视差图评价方式，其指标在像素梯度较高的区域有更大的权重，起到对视差图细节信息更关注的作用。

**关键词：立体匹配；视差；帧间相关性；代价函数；视差图评价**

# 目录

# 一、引言

## 1.1 课题背景及意义

用机器来模仿、重现人类视觉的处理的功能一直是计算机视觉领域的备受关注的研究热点。其中，在立体视觉的研究中，获取空间中的三维距离信息在无人驾驶、场景重建、机器人、虚拟现实等很多场景下都有着广泛的应用，便捷了人们的生活。

实现距离的计算或测量有不同的方法。主动测距（程距法）使用了专门设计的光源对物体进行照射，由反射信息计算获得距离。这种方法精度较高，但对环境产生影响，成本和对设备要求高。被动测距中，单目视觉试图通过统计假设来推断距离，但由于单一图片信息局限性，难度非常大。因此，双目等多视觉几何方法较为主流。

双目视觉是计算机视觉中物理上非常接近人眼处理的一个分支。仅通过一组相机被动获取自然光中的信息，通过几何关系获取视差变化，再使用一系列平滑算法来获取三维空间中的距离信息。这个过程与两个不同视角的人眼观察世界并处理信号的方式结构上非常类似。人的两眼所看到的图像有着一定的不同，不同距离的物体在眼睛中成像的位置差（即视差）不同，因此经过大脑的快速处理，人眼可即时分辨出物体的距离。

双目立体视觉使用双目摄像头模拟人眼，在实际应用中有着低成本、灵活度高、适应性强等特点。面对噪声、遮挡等不完美问题，人们研究出许多算法来对抗这些干扰。然而在处理尤其是高像素的图像时，一系列的视差计算、进行基于能量最小化的平滑处理是计算量巨大的，这在实际应用时是非常严峻的问题。如何保证高精度地高效计算是双目立体视觉的重要问题和挑战。

## 1.2 研究概述

关于立体匹配的方法历年来非常多。基于双目视觉的半全局立体匹配相关的工作中，主要的方向包括窗口匹配代价的设计、能量最小化的近似解获得、以及一致性检验等处理。

Birchfield和Tomasi在1998年提出了基于两窗口最大值最小值区间的匹配代价BT法[4]，计算简单效率较高。 Zabih和Woodfill提出了用于匹配代价计算的Census变换[5]，计算Census变换后窗口之间的汉明距离来定义匹配度，基于像素值之间的相对关系而不是绝对的灰度值，有着很好的鲁棒性。Žbontar 和Lecun在2015年提出通过训练CNN神经网络来计算匹配代价[6]。

仅在局部找到最佳匹配会带来偶然出现的误匹配，引起噪声的出现。为了达到平滑噪声的目的，将代价最小化（能量最小化）函数加入平滑项。理想情况下希望获得全局的最优解，然而这是一个NP完全问题。Kolmogorov和Zabih于2001年提出将图像的像素看作图的节点进行赋予标签，用图分割的方法解决能量最小化的问题[7]。Sun等提出使用置信度传播[8]，建立马尔可夫场来求解匹配问题。Hirschmuller提出半全局立体匹配方法[2]，聚合几条通向一点的路径上所有点的代价来近似全局，有着很好的效果。近几年也有许多基于深度学习的立体匹配方法如FlowNet[9]和DispNet[10]。

针对视频序列，M Li和L Shi等提出基于帧间关系的SGM计算加速DDR-SGM[1]，在精度影响较小的情况下大大提升了视频立体匹配计算效率。本作业主要重现了该工作的DDR-SGM部分，并讨论了新的匹配代价和视差图评价。

## 1.3 报告内容与结构

本文第一部分介绍了本作业课题的基本背景和领域研究现状。

第二部分介绍了双目立体匹配物理上的几何原理，介绍了对算法所计算出的视差图评价的体系。

第三部分是对半全局双目立体匹配的详细分步骤介绍，包括不同的匹配代价、针对噪声的基于半全局能量最小化的平滑方法、针对遮挡的左右一致性检查。并提出了一种将像素点灰度值分段对比的代价函数和一种关注细节的视差图评价方法。最后用实验对比了不同窗口大小、不同代价函数、不同聚合路径数的计算效率和精度。

第四部分是对视频序列的双目立体匹配方法介绍，包括基于动态视差搜索范围的加速方法和时间上的额外聚合路径。最后，对实现结果进行了展示。

第五部分是对基于深度学习的双目立体匹配方法的综述

第六部分对本作业主要内容进行了总结，对未来可进一步研究的方向进行了展望，

# 二、立体视觉几何与评价

## 2.1 双目视觉中的视差与深度信息

双目立体视觉的基本原理是，双目摄像机所拍摄的画面因摄像机位置不同而有一定的差别，相同的物点在图像中的位置会有一定偏差。该偏差即视差，与物体离相机的距离有着反比的关系。因此稠密视差图的方法中，可以用过视差的计算推得全图所有像素的距离信息。

由于双目摄像机物理设定的原因，获得的两幅图像之间存在着极线几何的约束条件。特别地，当两个相机轴线平行时称为平行立体视觉系统。这样的系统为视差的计算提供了很大的便捷。

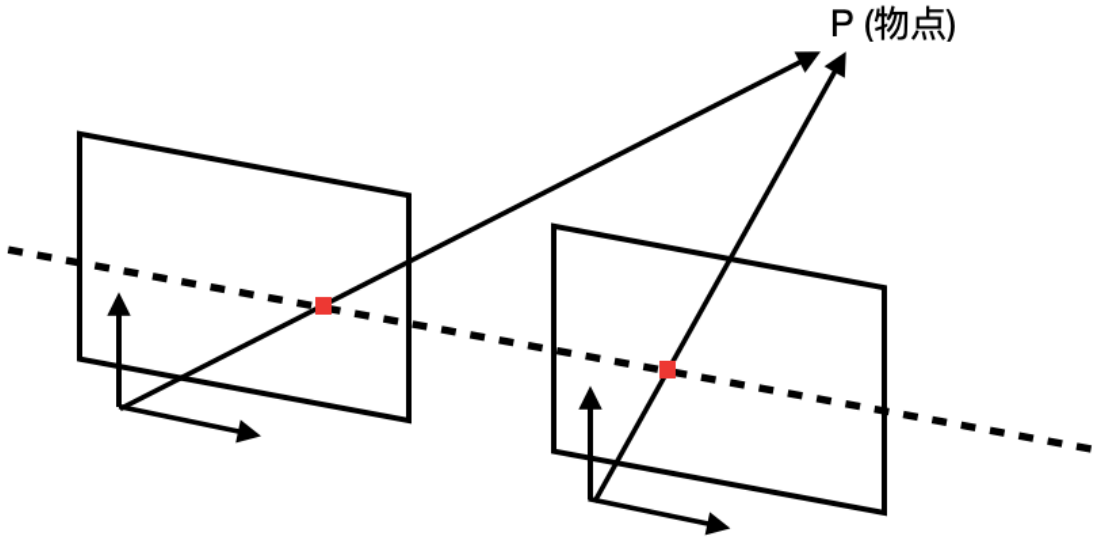



图 2.1.1 平行立体视觉系统

平行立体视觉系统中，一个物点P在两相机的画面中位于同一条水平线上，在匹配两图中的同名点时，仅需沿一行像素搜索，非常方便计算。通过某种匹配代价找到物点在两图像上的位置后，讲两位置坐标相减可得视差。如图2.1.2所示，可通过视差以及已知的相机参数恢复出P的距离。

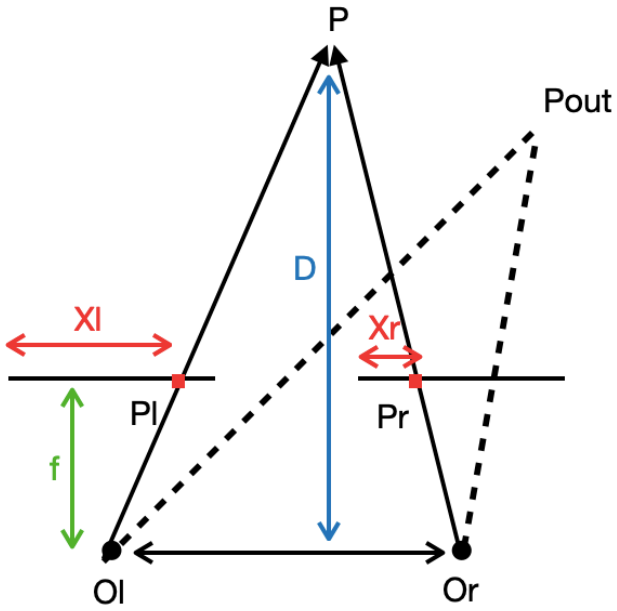



图 2.1.2 视差与距离的几何关系

图中两条等长共线线段和 $O_l$、$O_r$ 为双目相机；$P$、$P_{out}$为世界中的物点；$B$为基线距离；$X_r - X_l$ 为视差；$f$为相机的焦距；$D$为$P$到基线的距离。则通过三角关系可得，待求的$D$ 由下式给出：

$$D = \frac{Bf}{X_r - X_l}$$

因此，计算得视差即可用简单的计算获得距离。值得注意的是，如当物点位于$P_{out}$ 的位置时，无法在其中一个相机中成像，该区域理论上无法完成匹配。

## 2.2 立体匹配算法评价标准

本作业使用的单帧图像数据集和评价标准来自Middlebury。数据集中包括经过矫正的双目图像和实际测量得的ground truth。主要使用的评价标准是bad2.0和bad4.0。bad2.0计算方法是将算法所得的视差图逐像素与

ground truth视差图对比，误差超过两个像素的点占全部点的百分比。bad4.0同理。

在实际测试时发现，这些评价标准的缺点是对细节没有强调。算法如果忽视距离上的纹理但结果大致准确也可获得很高的分数。这对于需要精确细节的任务来说不足够理想。本作业在3.5中提出了一种关注细节的评价方法。

# 三、半全局双目立体匹配

## 3.1 半全局双目立体匹配(SGM)算法综述

单帧双目图像的半全局立体匹配算法步骤主要包括的：同名像点匹配、代价聚合、视差细化。

同名像点匹配的原理是对于基准图上的每一个点，另一张图的极线上寻找与之邻最相似的点。这个过程中需要用一定大小的窗口滑动对比检测。定义个邻域相似度的代价，可以获得左右两图每一对点的匹配代价。

由于这种方法找到的是窗口内的局部性最优解，在全局上看会存在偶然误匹配带来的诸多噪声。为了解决该问题引入带有平滑项的全局能量函数。该函数由原代价、第一惩罚项和第二惩罚项组成。第一个惩罚项用于保证连续平面的视差平滑变化，第二个惩罚项用于一定程度上允许物体边缘的视差突变。理想情况是求得该函数的全局最优解，但这是一个NP完全问题，因此希望通过半全局最优的方法近似。SGM方法中，对于每一个点，讲其自身匹配代价和沿着图像上多条路径累加后的代价求和，再求其最小值。这样既不需要对于每个点计算全图，也一定程度上考虑到了图像其他位置的能量，平衡了计算量和精度。最后，用胜者为王策略计算图像所有点的最佳视差值，获得平滑后的视差图。

由于双目视觉系统在物理上存在遮挡问题，有些点是理论上无法正确匹配的。针对这些点，有一系列视差细化的方法。例如左右一致性检验，将以左图为基准计算得的视差图与以右图为基准计算得的视差图比较，将视差值相差1以上的的像素点剔除，获得检验过的视差图。

## 3.2 代价计算

### 3.2.1 BT代价

BT代价由Birchfield和Tomasi在1998年提出[4]。其原理是，计算左图窗口的最大灰度值 $I_{pmax}$ 和最小像素值$I_{pmin}$，若右图窗口中心像素点像素值 $I_q \in [I_{pmin}, I_{pmax}]$，则 $cost1 = 0$，若不满足则$cost1$为$I_q$到$[I_{pmin}, I_{pmax}]$区间的最短距离。左右窗口关系互换计算得$cost2$。最终两窗口的匹配代价为:

$$cost(Np, Nq) = \frac{cost1 + cost2}{2}$$

### 3.2.2 Census代价

Census变换法由Zabih和Woodfill提出[5]，计算Census变换后窗口之间的汉明距离来定义匹配度。

Census变换将一个存有灰度值的窗口，即二维矩阵，变换为一个比特串。将窗口内每一个像素的灰度值$I_{N_p}$与窗口中心点灰度值$I_p$作比较，若 $I_{N_p} > I_p$ 则比特串后添加一位1，反之则添加一位0。这样窗口滑动过整个图片后，每一个像素点都对应一个Census变换比特串。两图希望匹配的点之间计算Census比特串的汉明距离（不同位的个数），归一化得Census变换的代价值。

该方法基于像素值之间的相对关系而不是绝对的灰度值，有着很好的鲁棒性；同时Census变换、汉明距离运算简单，计算上高效。

### 3.2.3 邻域分段代价

本作业中提出一种新的匹配代价计算方式。考虑到Census代价仅保留窗口内邻域点与中心点的大小关系， 希望设计一种在不过分增大计算量的条件下，能保留更多信息的代价函数，从而获得精度更高的匹配。

邻域分段代价的基本思路是，根据尝试经验预先设定一个阈值$R_{Np}$，对于图像中每一个点$p$，将像素的灰度值大小的一维空间分为四个区间，编号为A, B, C, D:

$$\begin{cases} A:[-I_{min}, I_p - R_{Np}] \\ B:(I_p - R_{Np}, I_p] \\ C:(I_p, I_p + R_{Np}] \\ D:(I_p + R_{Np}, I_{max}] \end{cases}$$

其中$I_p$是窗口的中心像素灰度值。$p$点邻域窗口内的每一个点都会落入四个区间中的一个。对右图所需匹配的$q$点进行同样的分段操作，$q$点邻域的的所有点也都会落入$q$的四个区间之一。

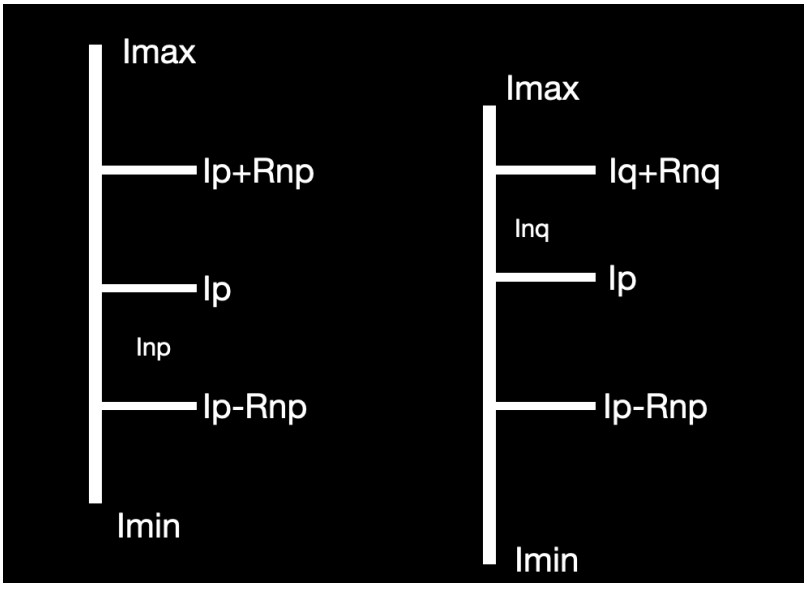



图 3.2.3 a 邻域分段代价示意图

对于窗口内的每一个位置的点，对比p窗口该位置点所处的区间和q窗口所处的区间，若两个区间编号相同则代价为0，若编号相邻则代价为较小的$cost_s$。除此之外，代价为较大的$cost_l$。特别地，阈值$R_{Np}$设为无穷或0时，邻域分段等同于Census变换。

算法伪代码如下：

**Algorithm 1 Divided Section Cost**

```
//win_size ： length of the side of window
//NBrange： range of divsion
//P_neighbors： Patch around P
//Q_neighbors： Patch arounf Q
//L_center： Gray value of P
//R_center: Gray value of Q
r = (win_size-1)/2;
sum = 0;
P_up = L_center+NBrange;
P_low = L_center-NBrange;
Q_up = R_center+NBrange;
Q_low = R_center-NBrange;
for m = 1:win_size
   for n = 1:win_size
      if (Q_neighbors(m,n) <= Q_low && P_neighbors(m,n) <= P_low) ||
(Q_neighbors(m,n) > Q_up && P_neighbors(m,n) > P_up) ||
(Q_neighbors(m,n) < Q_up && Q_neighbors(m,n) > Q_low &&
P_neighbors(m,n) < P_up && P_neighbors(m,n) > P_low)
      elseif (Q_neighbors(m,n) <= Q_low && P_neighbors(m,n) < P_up) ||
(Q_neighbors(m,n) > Q_low && P_neighbors(m,n) >= P_up) ||
(P_neighbors(m,n) <= P_low && Q_neighbors(m,n) < Q_up) ||
(P_neighbors(m,n) > P_low && Q_neighbors(m,n) >= Q_up)
         sum = sum + 1;
      else
         sum = sum +2;
      end if
```

```
    end for
end for
DSCost = sum/(win_size*win_size);
```

这种代价函数相比Census更耗时，精度也更高。具体实验结果和分析在3.6中阐述。

## 3.3 代价聚合

为了抑制偶然的误匹配所产生解决噪声，引入带有平滑项的全局能量函数:

$$E(D) = \sum_{p} (C(p, D_p) + \sum_{q} P_1 T[|D_p - D_q| = 1]) + \sum_{q} P_2 T[|D_p - D_q| > 1]$$

其中，$p$是原来的点，$q$ 是 $p$ 邻域内的点；$D_p$ 、$D_p$ 分别是 $p$点和$q$点的视差。$P_1$ 是第一个惩罚项系数，用于惩罚视差与$p$点差1的点；$P_2$ 是第二个惩罚项系数，用于惩罚视差与$p$点大于1的点。需要两个惩罚项是因为既需要保证连续平面内视差平滑变化，也需要允许物体边缘的视差突变。在此匹配问题化作能量最小化问题，希望求得上述能量函数的全局最小值。这是一个NP完全问题，计算成本过于高昂，因此SGM方法提出只寻求一个半全局的解在近似全局。

半全局的方法中，每一个点的能量函数不包含图像的全部点搜索，只累加某几条特定路径上的点的能量函数。

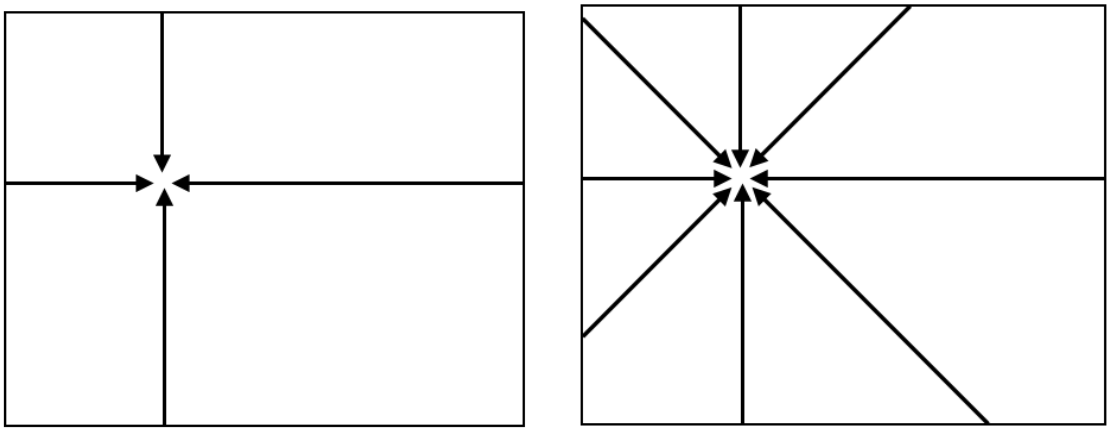


图 3.3.1 四路径代价聚合与八路径代价聚合

如图3.3.1所示，半全局的立体匹配方法中，每一个像素点的能量函数包函该点的在视差为$d$匹配代价和沿着路径方向相邻的点的能量函数的聚合，每条路径用动态规划的方法求解。计算时从箭头尾的边界开始，沿路径聚合至该点，过程中存储了路径上所有点在该路径上的代价聚合值，节约了计算效率。

$p$点沿某一条路径$r$进行代价聚合的能量函数如下:

$$L_r(p,d)=C(p,d)+\min\begin{cases}L_r(p-r,d)\\ L_r(p-r,d-1)+P_1\\ L_r(p-r,d+1)+P_1\\ \min_i L_r(p-r,i)+P_2\end{cases}-\min_i L_r(p-r,i)$$

其中 $C(p,d)$ 是原始匹配代价，$L_r(p-r,d)$、$L_r(p-r,d-1)$、$L_r(p-r,d+1)$在聚合路径上相邻点，视差为$d$、$d-1$和$d+1$的该路径能量函数；$\min L_r(p-r,i)$是该路径上能量函数最小值。对于相邻视差与原始点视差差1的情况加上惩罚项$P_1$，也允许该项取路径上的最小值加上惩罚项$P_2$。为了防止最后数值爆炸，式子最后减去前面项所能取的最小值。

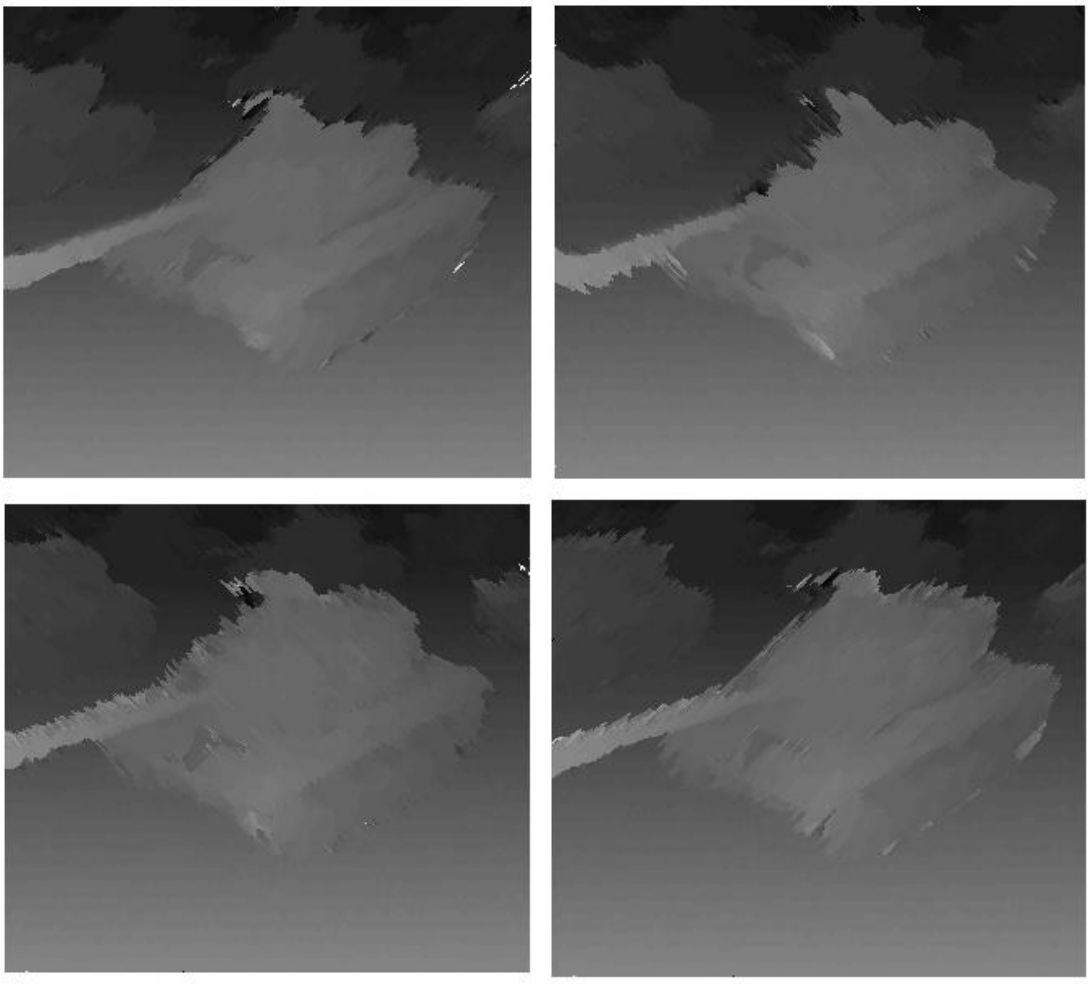


图 3.3.2 沿不同方向路径进行代价聚合的效果

对于每个点计算每条路径上的聚合值之后相加得到该点的能量函数值:

$$E(p,d)=\sum_{r}Lr(p,d)$$

再用winner takes all策略选取最佳视差值:

$$d*(p)=argmin_{d}E(p,d)$$

计算图像所有点的最佳视差值，获得视差图。

## 3.4 L-R check

针对光照条件、弱纹理、遮挡等问题有一系列视差细化算法，本作业中主要实现了左右一致性检测，即 L-R check。

由于角度问题，有些点只出现在了双目摄像机所拍摄的一张图片中，这样的点无法利用视差计算距离，应当被剔除。检查是否是遮挡点的方法是左右一致性检验。将以左图为基准计算得的视差图与以右图为基准计算得的视差图比较，将视差值相差1以上的的像素点的视差置为无效，获得检验过的视差图:

$$D_p = \begin{cases} D_p \ |d_p - d_q| \le 1 \\ o \ \ else \end{cases}$$

## 3.5 强调细节的评价标准

Middlebury所给出的bad1.0、bad2.0、bad4.0等系列评估方法对图像所有像素“一视同仁”，对待纹理强的和弱的部分代价权重是相同的。这样会导致有些视差计算结果在指标上不错，但视觉上丢失细节，而有时这种纹理细节往往是很关键的。对此，本作业一种新的视差图计算评估方法，设计了一种带有权重的评估方式。用一个窗口扫描ground truth视差图，获得每个点与邻域间变化大小的信息，邻域内变化越明显，给该点赋予权重越大。最后计算误差时，邻域间变化大的像素有更大的权重，因而评估更关注纹理复杂的细节。

评价算法伪代码如下：

**Algorithm 2 Differential Weighed Score**

```
//GT: disparity ground truth
//output: disparity map generated by an algorithm
//win_size: length of side of window
[r,c] = size(output);
[r_GT,c_GT] = size(GT);
radius = (win_size-1)/2;
weight_mask = zeros(r,c);
score = 0;
for i  = 1+radius:r-radius
   for j = 1+radius:c-radius
      GT_nb = GT(i-radius:i+radius,j-radius:j+radius);
      GT_diff = GT_nb - GT(i,j) ;
      weight_mask(i,j) = sum(abs(GT_diff(:))) +1;
      sum = sum + weight_mask(i,j)*abs(GT(i,j)-output(i,j));
   end for
end for
```

## 3.6实验结果和分析

本作业主要讨论了BT、Census和邻域分段代价。对比了不同窗口大小对BT代价和Census代价精度和计算时间的影响。

不同窗口的BT代价匹配得到的视差图如下所示：


图 3.6.1 不同窗口大小下的BT代价匹配视差图

| WIN SIZE | TIME | bad4.0 | bad2.0 |
|---|---|---|---|
| **3x3** | **48.3** | **0.294** | **0.304** |
| **5x5** | 51.9 | 0.322 | 0.333 |
| **7x7** | 59.7 | 0.342 | 0.353 |
| **9x9** | 60.7 | 0.353 | 0.364 |
| **11x11** | 58.7 | 0.363 | 0.374 |
| **13x13** | 63.5 | 0.377 | 0.388 |

图 3.6.2 不同窗口大小下的BT代价匹配的精度与计算时间

不同窗口的Census代价匹配得到的视差图如下所示：

**5x5**

**7x7**

9x9

11x11

13x13

图 3.6.3 不同窗口大小下的
Census代价匹配的精度与计算时间

| WIN SIZE | TIME | bad4.0 | bad2.0 |
|---|---|---|---|
| **5x5** | **36.3** | 0.213 | 0.219 |
| **7x7** | 46.3 | 0.205 | 0.210 |
| **9x9** | 49.4 | 0.200 | 0.202 |
| **11x11** | 56.3 | 0.190 | 0.194 |
| **13x13** | 61.6 | **0.185** | **0.189** |

图 3.6.4 不同窗口大小下的Census代价匹配的精度与计算时间

BT和Census都体现出窗口越大计算时间越长。这个结果与预想相符，因为窗口越大，总共需要遍历的像素数几乎不变，但每个窗口内迭代次数就越多，计算时间越长。另外试验结果显示，Census窗口越大精度越高，视觉上也越平滑。

在books数据集上，邻域分段代价与Census的比较如下：

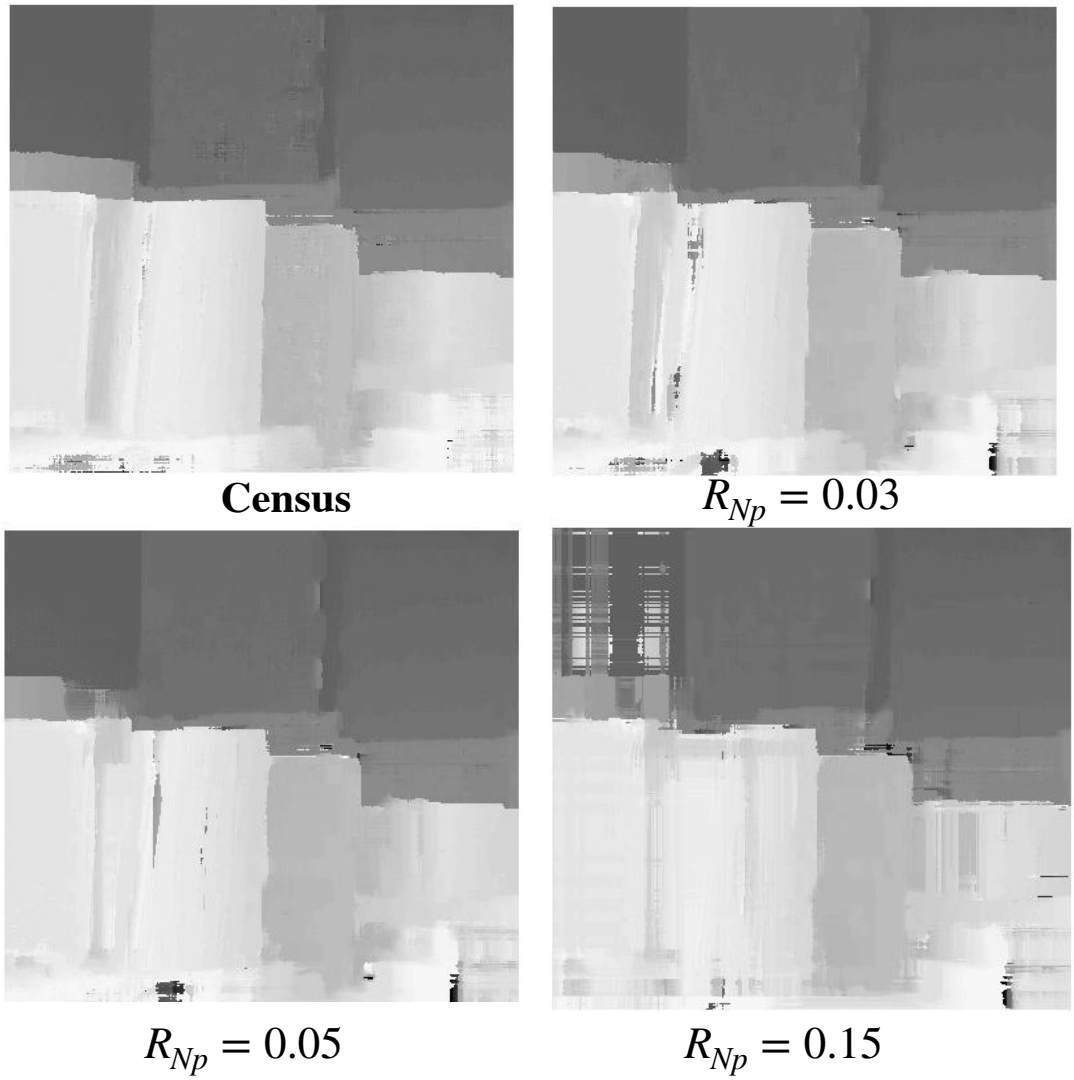



图 3.6.5 不同阈值的邻域分段代价与Census的匹配视差图比较

| WIN SIZE | TIME | bad4.0 | bad2.0 |
|---|---|---|---|
| **Census** | **60.1** | 0.127 | 0.132 |
| **0.03DS** | 88.1 | **0.106** | **0.111** |

图 3.6.6 $R_{Np}$=0.03邻域分段代价与Census的匹配精度和计算时间

由上表可知，邻域分段代价在实际应用中，计算时间大于census，精度略高于Census。

代价聚合部分，本作业讨论了如图3.6.7所示两种4路径聚合和一种8路径聚合。

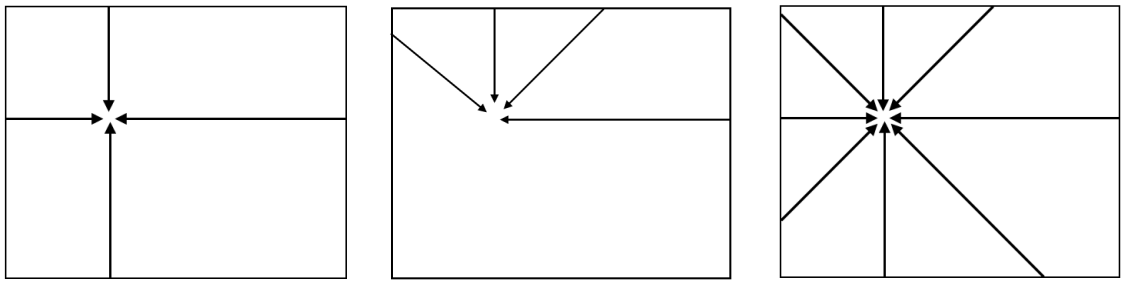


图 3.6.7 4路径聚合A、4路径聚合B和8路径聚合

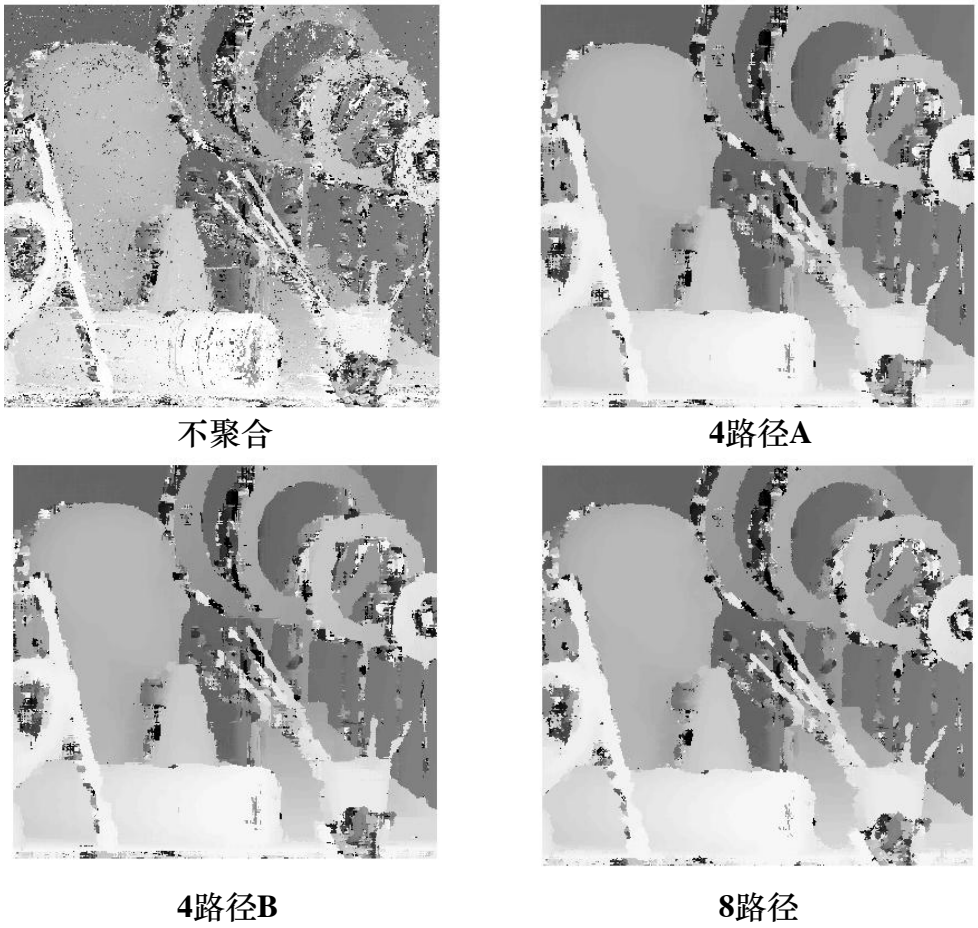



图 3.6.8 不聚合、4路径聚合A、4路径聚合B和8路径聚合的匹配效果

直观视觉上4路径聚合得到的视差图明显比不聚合的更加平滑。不同方向的4路径与8路径聚合肉眼很难看出区别。但放大局部细节可发现8路径聚合较好地处理掉了一些4路径聚合遗留下的噪声点，如图3.6.9所示：

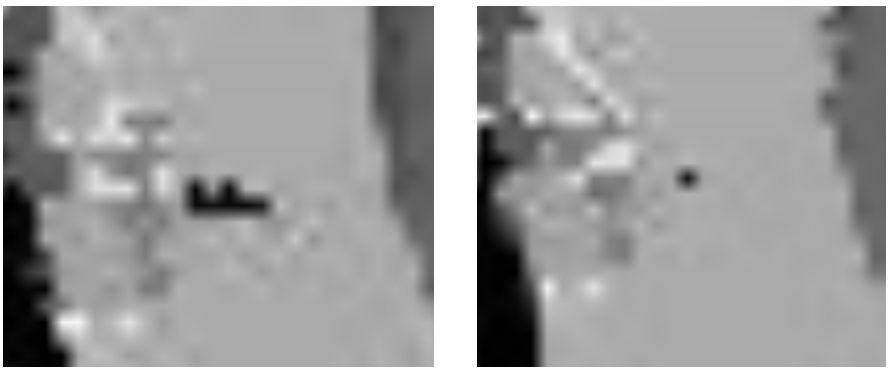



图 3.6.9 4路径聚合与8路径聚合结果细节放大

| PATH | TIME | bad4.0 | bad2.0 |
|---|---|---|---|
| NO CA | **61.3** | 0.127 | 0.132 |
| 4PATH A | 106.8 | 0.085 | 0.089 |
| 4PATH B | 94.7 | 0.086 | 0.091 |
| 8PATH | 126.9 | **0.080** | **0.084** |

图 3.6.10 不同聚合方法的计算时间和精度

从图3.6.10中可知，相比不进行聚合，三种聚合方法都有着计算时间更长，精度更高点特点，符合对代价聚合的 作用定位。其中对称性强的4路径聚合A比B计算时间长，精度相差不多。8路径聚合最耗 时，精度最高但不显著，在算力有限的情况下性价比不及4路径聚合。

视差细化中实现了聚合后的L-R check，检查出了一定量的遮挡点。

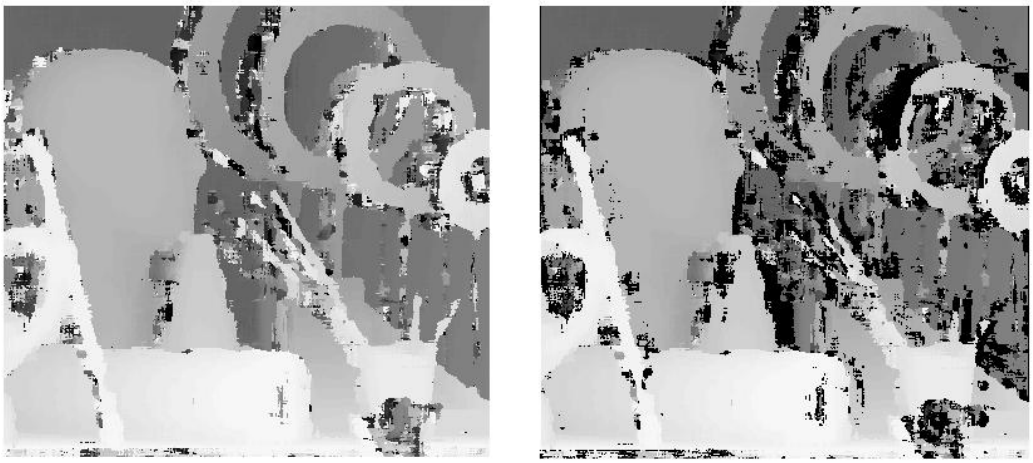


图 3.6.11 有无L-R check对比

# 四、视频序列的立体匹配

## 4.1 帧间相关性与动态视差范围

半全局立体匹配在计算上依然是较为耗时的。考虑到视频序列中，除画面突变之外，大部分的两帧之间并没有非常大的视差变化，因此在时间上是相关的，可以利用这种相关性简化匹配的计算。

DDR-SGM原作[1]中引入的动态视差范围是为了在视频序列的视差计算中提高计算效率。首先，预先设定一个搜索半径。第一帧按原SGM方法计算，从第二帧开始，视差的搜索不再是全范围，而是以上一帧该像素的视差为中心，搜索半径为半径的区间内。每帧以前一帧的视差信息为基础，在小于全范围的搜索范围进行搜索。

由于这样这样在边界等位置会有最佳视差在动态搜索范围之外的，引入一个与预设临界值对比判断。如果某像素点随后的聚合代价高过了临界值则意味着有可能最优的视差在搜索范围之外，因此对该点再进行完整全范围的SGM。

原文中实验结果显示在精度仅下降0.1的情况下，计算效率获得了大幅度的提升。

## 4.2 代价聚合时序路径

单帧图像包含的是二维的信息，可用一个二维矩阵来表示。代价聚合步骤的聚合路径则选在该平面上的不同方向。视频序列则可由一个三维矩阵来表示，每一个切面为一帧。

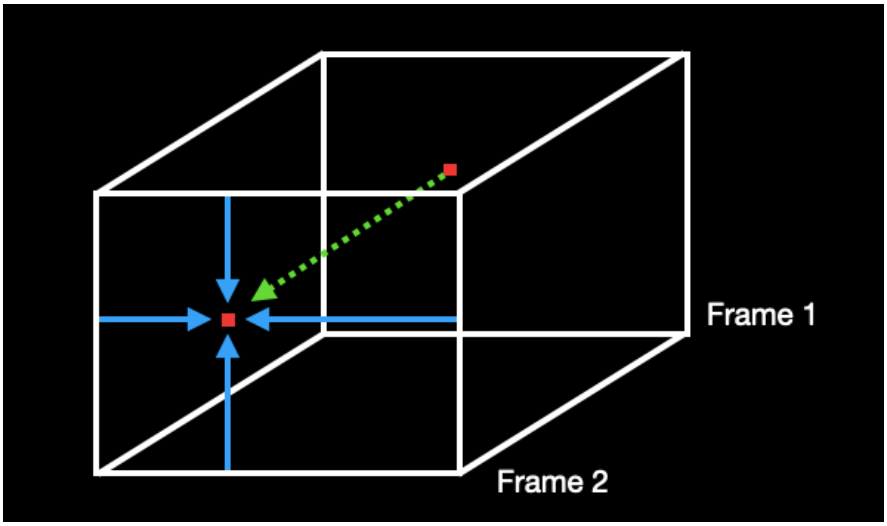
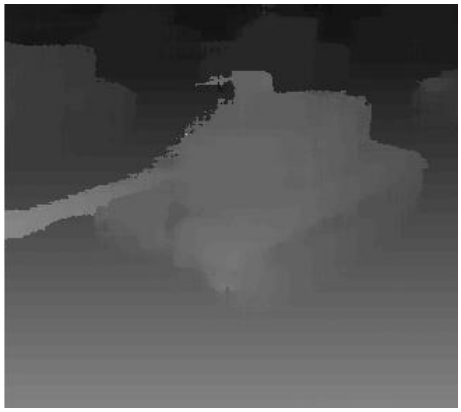



图 4.2.1 视频序列中时间维度的代价聚合

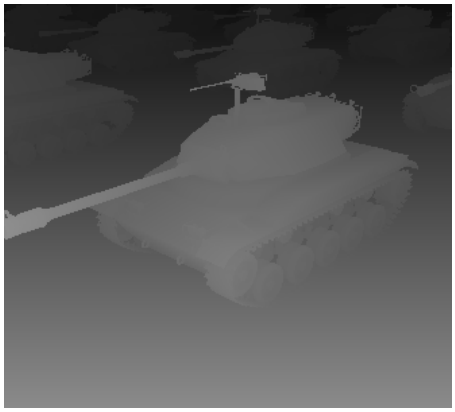
如图4.2.1中的绿色虚线所示。多了一个时间维度意味着多一条可以进行聚合的路径，可以对同一个像素在时间上也进行平滑。文章[1]引入了时间维度的代价聚合，也在本 工作中重新实现。

## 4.3 实验结果和分析

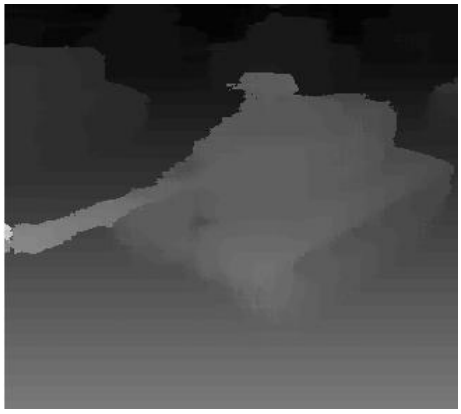
本重现工作中参数上选择了保守的较大搜索范围R= 40，聚合代价临界值TH=10。如图4.3.1所示，累积过程中出现个别像素滞留情况，精度随时间有下降趋势。本文使用的数据集是 Christian Richardt 搭建的视频序列立体匹配数据集[13]中的Tanks。



**F1 DDR-SGM**

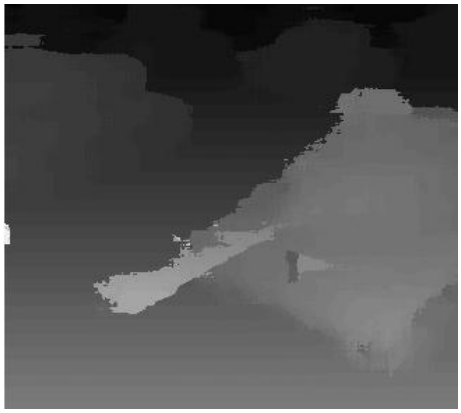


**F1 GT**

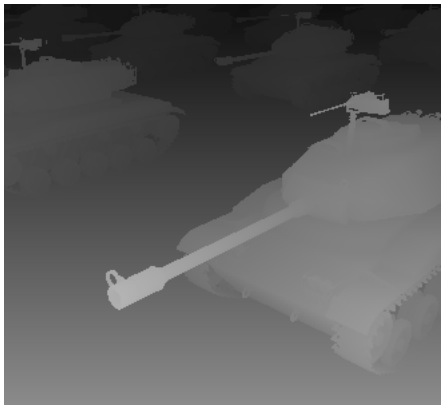


**F10 DDR-SGM**

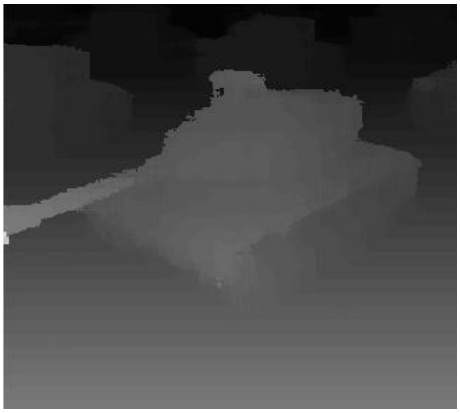
**F10 GT**



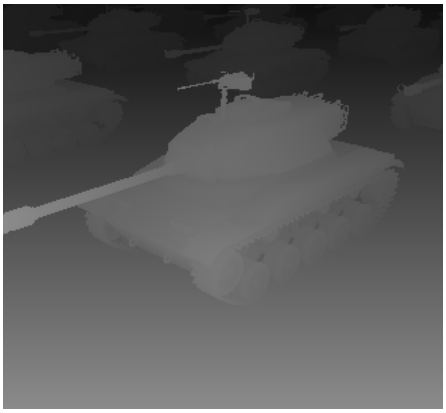
**F20 DDR-SGM**



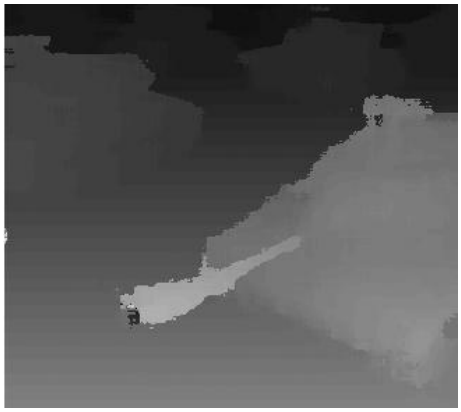
**F20 GT**



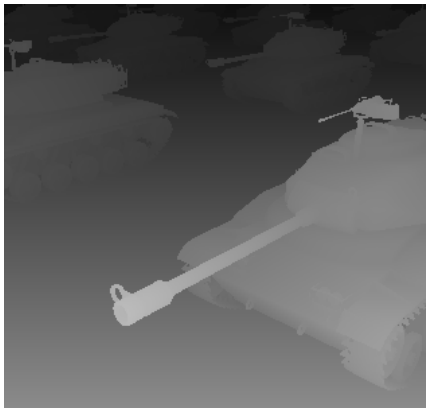
**F40 DDR-SGM**



**F40 GT**



**F70 DDR-SGM**



**F70 GT**

图 4.3.1 视频序列DDR-SGM计算结果与GT比较

对比每帧都使用SGM和使用DDR-SGM的精度和计算时间：

| METHOD | TIME | avrbad4.0 | avrbad2.0 |
|---|---|---|---|
| SGM | **70.8** | 0.022 | 0.156 |
| DDR-SGM | 55.8 | **0.028** | **0.184** |

图 4.3.2 视频序列DDR-SGM与SGM的计算时间与精度

由上表可得DDR-SGM相比SGM精度略有下降，计算效率有较大提高。

# 五、基于深度学习的立体匹配综述

近年来，随着深度学习的逐渐兴起，基于深度神经网络的立体匹配算法也层出不穷。

立体匹配的第一阶段，也是非常关键的一个步骤是匹配代价的选择。如何定义两个patch之间的相似性是该部分的关键问题。具有代表性的工作是Zˇbontar 和Lecun在2015年提出的MC-CNN[6]。通过训练卷积神经网络对图像patch之间的相似度进行学习。从神经网络功能分类上，属于一个二分类问题。通过监督学习的方法，提供给CNN带有“相似”或“不相似”标签的patch对。训练完成后的CNN作为代价函数，可以根据未见过的输入patch对来给出二者相似度的判断，作为输出。代价函数设计完成后，MC-CNN后续步骤选择较为典型，使用了cross-base代价聚合、半全局匹配。视差细化步骤进行了左右一致性检查，亚像素级增强、中值滤波、双边滤波等操作。结果上看，MC-CNN算法在KITTI2012、KITTI2015和Middlebury 数据集上表现比传统方法好，速度也有提升。

由于MC-CNN依然需要对patch对patch对比，如果N是每行patch的总数，该算法匹配部分时间复杂度O($N^2$)。为了降低计算时间成本，Luo[11]等人提出将双目匹配化作一个多分类问题，在每一个像素点做分类，每一个class就是一个视差值。以将左图为基准为例。将左图中待匹配的patch输入到一个卷积神经网络中，前向传播得到一个长度为64的左图特征volume。将右图一整行的图像输入一个卷积神经网络，前向传播获得一个大小为64x视差范围的右图特征volume。讲两个特征volume作内积，通过softmax分类得到不同视差值的概率。计算的代价函数使用了交叉熵。最终结果上看误差略大于MC-CNN，但由于每张图只需要前向传播一次，计算速度有显著的提升。

Shaked和Wolf提出了一种基于多级加权残差的跳层连接的Highway Network架构[16]来计算匹配代价和一个全局的视差网络去预测视差置信度得分，这方便进一步优化视差图。该方法可计算每一个可能的视差的匹配误差，通过复合损失函数，支持patch的多级比较。引入了一个新的视差细

化处理，用一个卷积神经网络池化多个视差值来获得全局信息，最后输出预测视差图和置信度。新的置信度精细化步骤更好地检测了异常点。

立体匹配的后处理阶段也有许多工作引入了深度学习的方法。Guney和Geiger在Displets[9]的工作中考虑到现实世界中的物体一般结构规则，不会随意变化，因此关注了语义信息在图像分割中的作用。该工作中，使用SLIC将图像分割为超像素平面，采样了CAD三维结构（凸面车等），局部平面和视差图匹配获得平面参数。构建了一个新的能量函数，包括数据项、局部平滑项和特殊的“Displet Potentials”项。数据项是由稀疏视差图所得的惩罚偏差；局部平滑项与传统SGM类似，惩罚了视差不连续性；Displet Potentials是标识一个符合特定语义类的可能的几何区域，其一元可能性用来描述图像中形状符合特定目标类的区域被指定给语义类标签。该方法利用了语义信息，在减少弱纹理和反射区域的错误匹配上有着良好的表现。

考虑到传统SGM的惩罚项系数P1和P2需要根据图像人工调参的问题，Akihito和Marc提出将SGM与神经网络结合，使用深度学习自动学习到惩罚项参数。训练过程中，希望最小化路径代价和邻域代价；测试阶段，SGM依靠SGM-Net给出的P1、P2进行惩罚。

MC-CNN等方法对神经网络的使用仅限于匹配代价的构造，后续的平滑和视差细化仍使用传统方法。为了不进行后处理进而实现端到端，渐渐出现了一些完全用神经网络预测视差的。这种端到端架构的启发之一是Dosovitskiy等提出用深度神经网络预测光流场的FLowNet[9]。光流是利用视频序列中相邻帧之间的相关性计算物体运动信息的方法，要求邻帧之间亮度恒定，物体运动连续，空间上有一致性。FlowNet分一个收缩卷积神经网络和一个扩大卷积神经网络。收缩部分的一个简单结构为FlowNetSimple，简单地将图片叠加在一起送入CNN中，从中提取出光流信息。另一种方法是网络独立提取两张图片的高层特征再进行混合，成为FlowNetCorr。扩大部分由上卷积层组成，将上采样对光流的预测和收缩部分所得的特征图相联系，不断提升分辨率。在四次提升分辨率操作后，再进行后处理已无明显提高，意味着效果已经达到。由于CNN的训练需要

大量的数据，该工作中创建了Flying Chairs数据集。数据集中有大量不同的椅子和不同角度，再加上图像位移，反转，方法，加高斯噪音，改变亮度，对比度，gamma值和颜色等数据扩充的防过拟合操作，为训练提供足够的有效数据。第一版的FlowNet计算速度虽快但准确度表现一般。后来的FlowNet 2.0 进一步扩充了数据集并复杂了训练策略，在性能上有了显著的提升。

DispNet[10]是Mayer等提出的基于FlowNet框架的修改，是一种应用神经网络学习计算视差的方法。相比FlowNet，DispNet在扩大部分的每个deconvolution层和前一预测结果的concat后，增加了一个卷积层使结果图更为平滑等。Tonmoy等在2019年提出的AutoDispNet[18] 使用现有的AutoML扩展其在Disp上面估计的能力。

立体匹配的内在的病态区域，如遮挡、重复特征、无纹理区域等难产生高质量的视差问题一直是该领域的关键难点。对此，Pang等人提出一种新颖的由两个阶段组成的堆叠卷积神经网络结构[19]。在拓展了DispNet的基础上引入了一种两级网络，称为级联残差学习CRL。在第一阶段利用DispNet的基础上加上反卷积模块。这样能使视差图获得更多细节。在第二个阶段结合第一阶段产生多尺度的残差信号，修正由第一阶段产生的初始视差。第二个阶段不是直接学习视差，而是通过残差学习提供更有效的精细化。这个思路与传统SGM有一定相似性，第一阶段计算视差，第二阶段细化。两个阶段的输出的和给出最终的视差。第一级和第二级网络分别计算视差图和多尺度的残差。两级网络的输出相加构成最终的视差图。该工作在多个尺度上引入残差学习机制，并通过ground-truth视差和初始视差直接监督，在视差精细化上有着非常优良的效果。

上述几个工作中，DispNet和CRL的共同点是利用了低层次和高层次的级联特征和多层信息。CRL还通过多层监督进一步精细视差的计算结果。

另一方面，深度学习中卷积这一操作是局域性很强的，感受野的有限性会限制算法获得全局的最优解。关于考虑全局的方法也是近年来的热门。Chang和Chen提出了PSMNet[20]，包含了一个用于结合全局的金字塔

池化模块SPP和重叠的用于匹配代价聚合的沙漏模块。SPP最早用于去除CNN的尺寸约束，后被广泛用于语义分割的问题。PSMNet中，将左右图分别输入相同权重的CNN计算得特征图，用SPP模块来结合像素环境信息，获取特征。串联不同尺寸的次级区域的表现和一个用于特征融合的卷积层。左右图像的特征被用于构成一个四维的匹配代价卷。最后通过一个三维卷积神经网络进行代价聚合。最后依然是通过softmax分类，得到每一个视差值的概率作为输出。

贯穿立体匹配研究历史的重要问题之一—计算速度问题上，谷歌2018年推出了推断速度高达60Fps的Stereonet[21]，是第一个实时双目深度估计网络。该工作仿照了经典立体匹配的计算步骤。通过左右图共享权值的Siamese 网络提取左右图特征，将特征图的差异作为基础得到代价volume。借鉴了Mobilenet，Squeezenet等小网络领域成果，将特征图显著减小，极大降低了网络计算负担。Leaky ReLU等操作也使得其可在极小的计算负担下获得更多的图像特征。文章亮点在于网络结构设计，保留了较小但特征丰富的代价volume。再用一种层次化的网络估计残差，利用残差和粗糙的视差图分层优化，得到保留边缘的视差图。

与主动测距结合方面，谷歌&普林斯顿的研究人员[14]在ECCV 2018上提出了一种主动双目立体成像系统的深度学习解决方案。这是第一个主动的端到端双目深度学习方法。由于缺乏ground truth，该工作的方法是自监督，但仍然达到了1/30像素级的精度，并很好地处理了过度平滑和边缘保留问题。工作中提出了一种基于局部对比度归一化(LCN)的重构代价，对噪声弱纹理和光强变化具有鲁棒性。并且，文中提出了基于窗口的权重自适应代价聚合方法来抑制局部最优解的问题。最后计算代价，检测并提出了出了遮挡点。

总的来说，立体匹配方法有着明显的朝深度学习方向趋势。这与神经网络强大的抽象和表达能力、训练完成后使用网络计算的快速等因素有关。深度学习方法在计算机视觉的多种领域都得到了很好的应用。然而端到端的深度学习方法的缺点依然是不可解释性，这一点上说明传统的双目

视觉方法依然有很多可进一步探索之处。立体匹配领域正在朝着实时、高精度的方向迈步。需要解决的依然是计算素和全局性问题。

## 六、总结与展望

本作业围绕双目立体匹配，对比了BT和Census代价在不同窗口下的视觉效果、精度和计算时间。对比了不同方向的四路径和八路径代价聚合的视觉效果、精度和计算时间。视差细化中实现了左右一致性检验。针对视频序列实现了时间维度上的代价聚合，实现了动态视差范围的SGM，对比了DDR-SGM与SGM的精度和计算时间。

提出了一种保留更多窗口信息的新的代价函数“邻域分段代价”，与Census进行了精度和计算时间的对比。

提出了一种新的视差图评价方式，在像素梯度高处有着更大权重，对视差图细节信息更关注。

未来可进一步工作的方向包括重视细节的评价的定量探究，DDR-SGM精度随时间变化的研究，基于颜色信息的视差计算等。

## 七、参考文献

[1] M.Li, L.Shi, X.Chen, etc. Using Temporal Correlation to Optimize Stereo Matching in Video Sequences[J].IEICE TRANS. INF. & SYST, E102–D(6):1183-1195, 2019.
[2] Hirschmuller H. Accurate and Efficient Stereo Processing by Semi Global Matching and Mutual Information. *CVPR* , 2005.
[3] Hirschmuller H. (2008). Stereo Processing by Semiglobal Matching and Mutual Information. *Pattern Analysis and Machine Intelligence* .
[4] Birchfield, Tomasi. Depth Discontinuities by Pixel-to-Pixel Stereo 1998 IEEE

[5] Zabih R. and Woodfill J. (1994) Non-parametric local transforms for computing visual correspondence. Computer Vision—ECCV'94,
[6] J. Zˇbontar and Y. Lecun, "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches," vol.17, pp.2287–2318, 2015.
[7] Kolmogorov V., Zabih R. Computing visual correspondence with occlusions using graph cuts. IEEE Int. Conf. Comput. Vis. 2 , 508–515, 2001.
[8] Jian Sun,Nan-Ning Zheng, Heung-Yeung Shum. Stereo matching using belief propagation.IEEE Transactions on Pattern Analysis and Machine Intelligence. 2003.
[9] Dosovitskiy, A, et al FlowNet Learning Optical Flow with Convolutional Networks in 2015 IEE International Conference on Computer Vision ICcV) 2015.
[10] Mayer, N, et alA Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation in 2016 IEEE Conference on Computer Vision and Patter Recognition (CVPR) 2016.
[11] J Luo, W.A.G. Schwing and R Urtasun: Efficient Deep Learning for Stereo Matching in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2016.
[12] Chen z, et al. A Deep Visual Correspondence Embedding Model for Stereo Matching Costs. in 2015 IEEE International Conference on Computer Vision (ICCV) 2015.
[13] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N.A. Dodgson, "Real-Time Spatiotemporal Stereo Matching Using the Dual-Cross-Bilateral Grid," Proc. Computer Vision - ECCV 2010, European Conference on Computer Vision, Heraklion, Crete, Greece, Sept. 5-11, 2010, pp.510–523, 2010.
[14] Y. Zhang et al. ActiveStereoNet: End-to-End Self-Supervised Learning for Active Stereo Systems. ECCV2018.
[15] F. Guney, A. Geiger: Displets:Resolving Stereo Ambiguities using Object Knowledge. CVPR
[16]Shaked A, Wolf L. Improved Stereo Matching with Constant Highway Networks and Reflective Confidence learning. CVPR 2017: 4641-4650.

[17] Akihito S, Marc P: SGM-Nets: Semi-global matching with neural networks. CVPR 2017.
[18] Tonmoy S et al. AutoDispNet: Improving Disparity Estimation With AutoML. CVPR 2019.
[19] J Pang, et al. Cascade Residual Learning: A Two-stage Convolutional Neural Network for Stereo Matching. CVPR 2017.
[20] JR Chang, YS Chen. Pyramid Stereo Matching Network. CVPR 2018.
[21] S Khamis et al. StereoNet: Guided Hierarchical Refinement for Real-Time Edge-Aware Depth Prediction. ECCV 2018.