Semi-Global Matching Algorithm Description

一、Census Calculation

1、保存结果

分别保存左图和右图的 Census 的计算结果。

census_l: width * height

census_r: width * height

2、Census 的计算

根据实际的需要选择不同的窗口大小，例如 5*5

0	1	2	3	4
5	6	7	8	9
10	11	C	12	13
14	15	16	17	18
19	20	21	22	23

其中，C 代表当前像素，数字代表 5*5 窗口内像素的索引。

计算规则：

if index_pixel < C result: 1

else result: 0

计算的结果一共有 24 位。

3、特殊情况

census_l 和 census_r 的前两行及后两行分别填充 0。

二、Hamming Distance

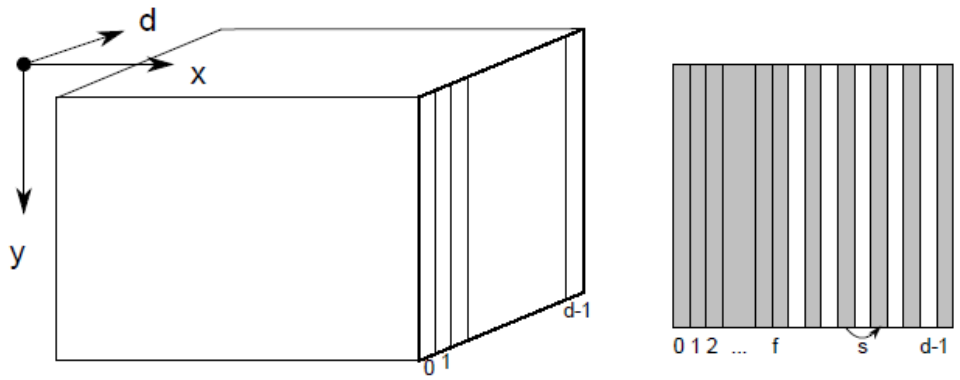
1、Hamming Distance 的计算

两个字符串对应位置的不同字符的个数

1011101　1001001　　HD_Result：2

2、保存计算的结果

dsi_cube：(Width*Height*DispCount）



3、建立视差立方体(Width*Height* DispCount)

dsi_cube(v, u, d) = census_l(v, u) HD census_r(v, u + d - DispCount + 1)

$$d = \{0,1,\cdots,\mathrm{DispCount}\}$$

4、特殊情况

图像的前两行及后两行的 dsi_cube 分别赋无效值。

三、动态路径规划

1、保存结果

S：(Width*Height*DispCount)

2、DP 计算

$$L_r\big((v,u),d\big)$$

$$= C\big((v,u),d\big)$$

$$+ \ min\big[L_r((v,u)_r,d), L_r((v,u)_r,d-1)$$

$$+ P_1, L_r((v,u)_r,d+1) + P_1, \overset{min}{i} L_r((v,u)_r,i) + P_2\big]$$

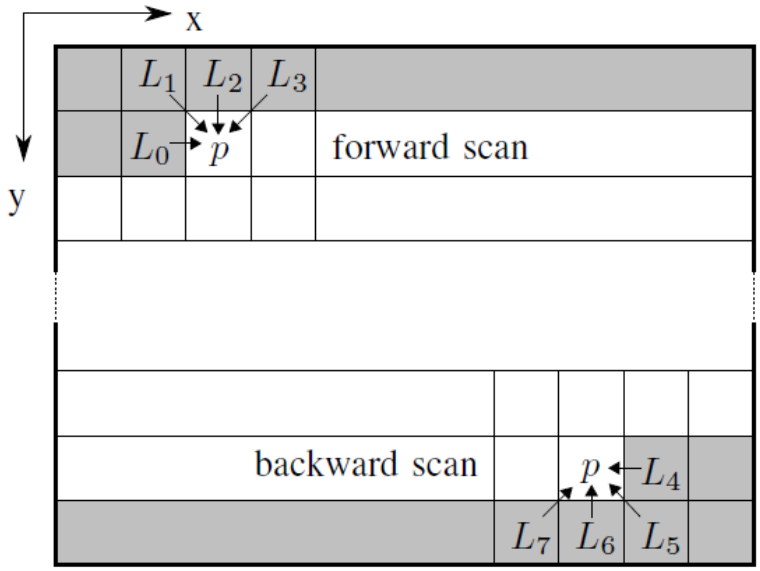
$$\mathrm{S}\big((\mathrm{v},\mathrm{u}),\mathrm{d}\big) = \sum_{\mathrm{r}} L_r\big((v,u),d\big)$$

3、DP 计算过程

实际的算法当中，选取 8 个方向。为了快速的计算，前四个方向从图像的第一个像素开始计算；后四个方向从图像的最后一个像素开始计算。

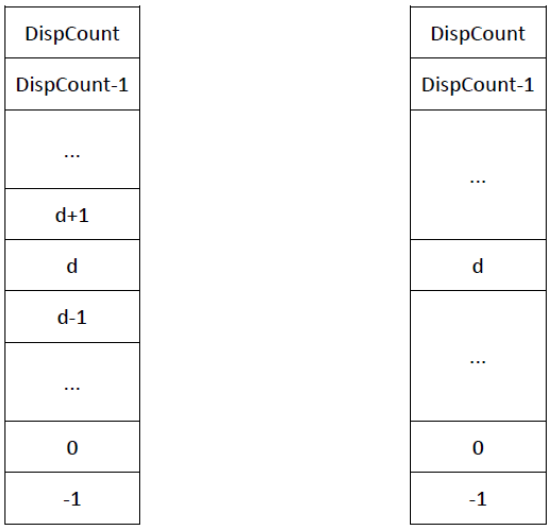
3.1 描述 8 个方向

算法中具体的 8 个方向如下图所示：



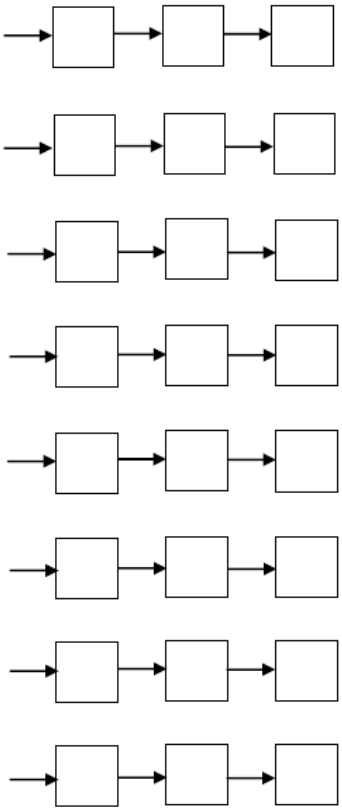
3.2 每个方向的计算过程

每一个方向的计算过程都是一样的，如下图所示：



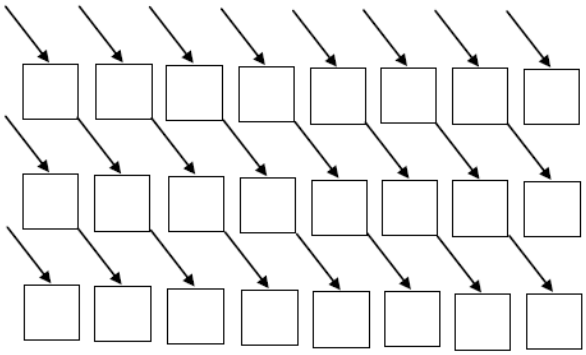
3.2.1 L_0 方向的表示

为了符合 S32V 的并行特性，L_0 方向如下图所示：



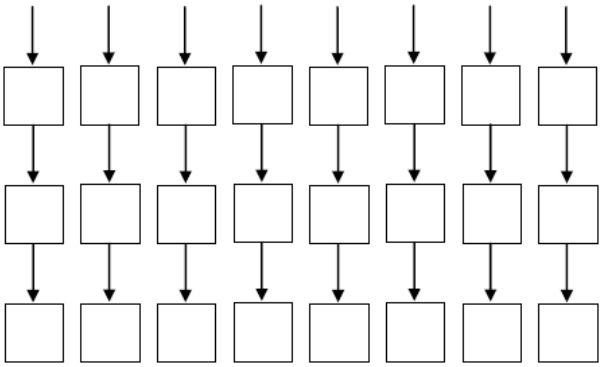
3.2.2 L_1 方向的表示

L_1 方向如下图所示：



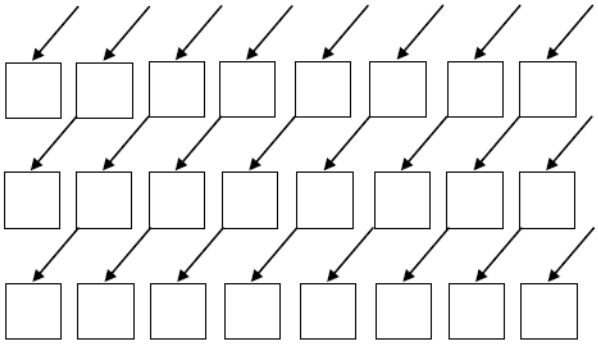
3.2.3 L_2 方向的表示

L_2 方向如下图所示：



3.2.4 L_3 方向的表示

L_3 方向如下图所示：



3.2.5 其他方向

其他的方向跟上图类似，但是要从图像的最后一个像素开始计算。

4、特殊情况

L_0 方向的情况下，第一列特殊，需要做特殊的处理；L_1、L_2、L_3 方向就是第一行比较特殊，需要做特殊处理。

四、WTA

当计算完全 DP 的时候，剩下的就是求 WTA，WTA 分为两个 WTA_L 和 WTA_R;

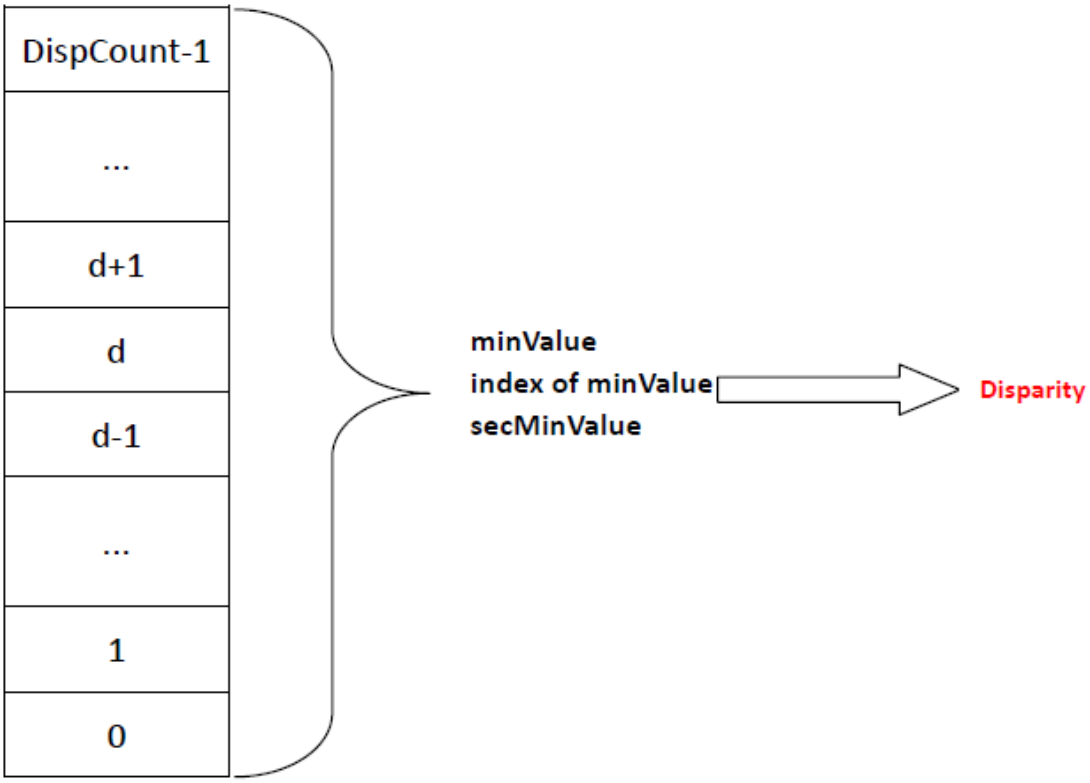
1、保存结果

WTA_L：(width*height)

WTA_R：(width*height)

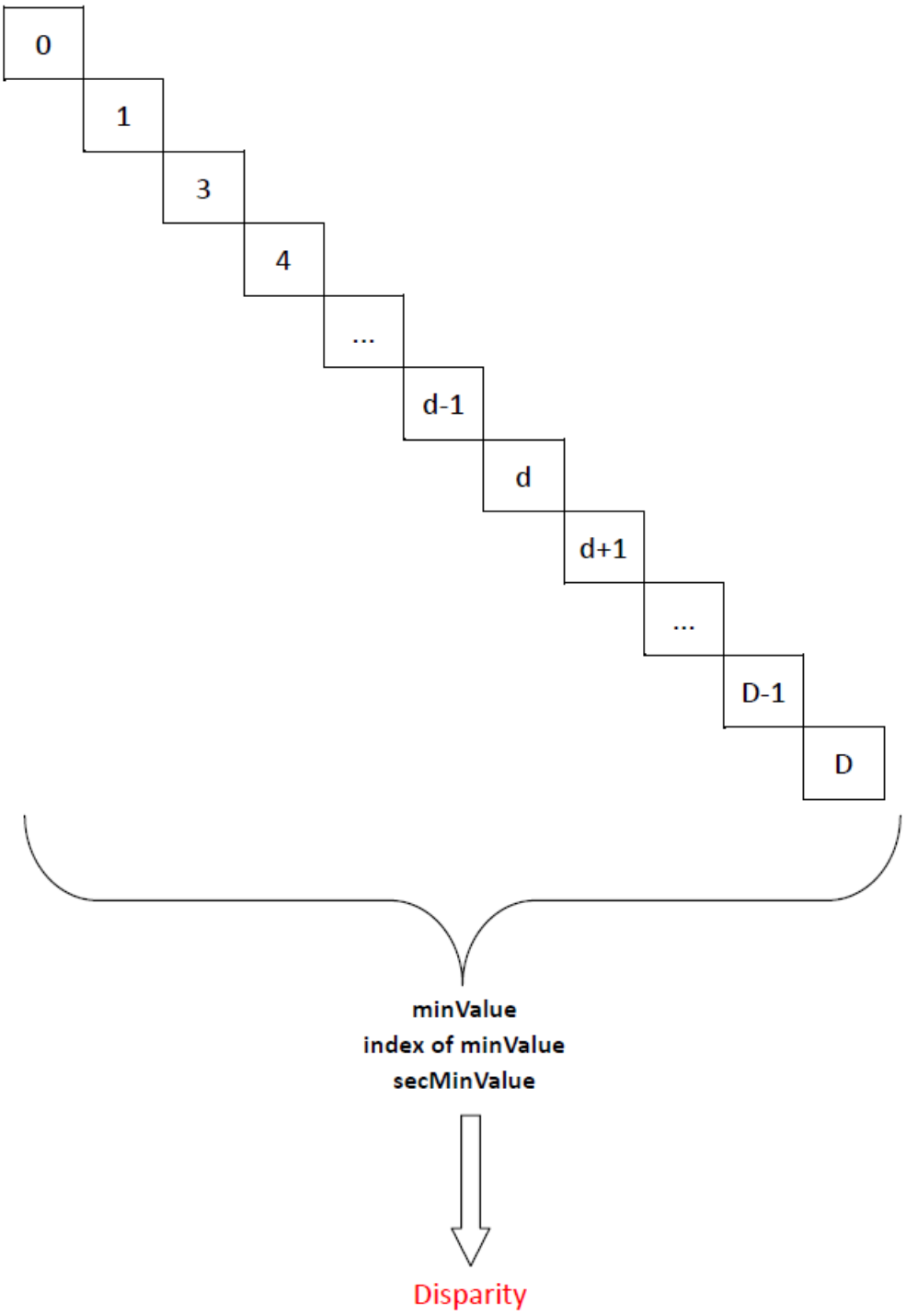
2、WTA_L 计算

WTA_L 的计算过程如下图所示：



3、WTA_R 计算

WTA_R 的输入和 WTA_A 一样，但是计算过程不一样。具体的 WTA_R 计算过程如下图所示：



其中，D 表示最大的视差，Disparity 表示 WTA_R 中某一个值。

五、Left-Right Check

1、保存结果

DispImg：(width*height)

2、计算过程

根据计算过程得到最终的视差，并将结果保存为 DispImg。