ISP-DEMOSAICKING（中）

食鱼者

202203

目 录 CONTENTS

01 算法原理精讲

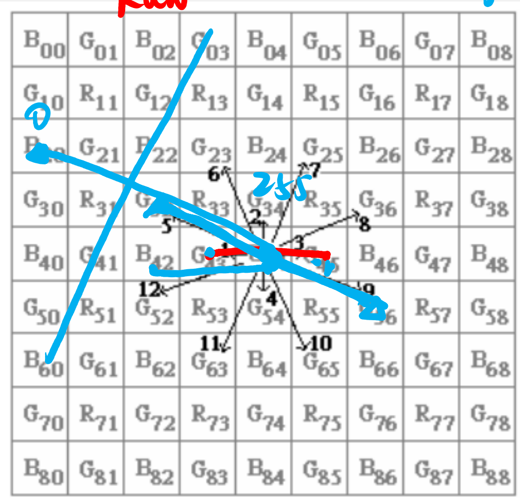
算法精讲-方向加权法

- RB缺失的G通道的插值

1.2.3.4 方向上 $k_n = 1$

5-12方向 $k_n = 0.5$ $\frac{1}{\sqrt{5}}$

Raw



n	v_n	h_n	n	v_n	h_n
1	0	-1	2	-1	0
3	0	+1	4	+1	0
5	-1	-2	6	-2	-1
7	-2	+1	8	-1	+2
9	+1	+2	10	+2	+1
11	+2	-1	12	+1	-2

$$I_n(i,j) = k_n\{abs[P(i+v_n, j+h_n) - P(i-v_n, j-h_n)] + abs[P(i+2v_n, j+2h_n) - P(i,j)]\}$$

$$k_n(|P(4,3) - P(4,5)| + |P(4,2) - P(4,4)|)$$

$$I_5(4,4) = k_n(|P(3,2) - P(5,6)| + |P(2,0) - P(4,4)|)$$

12个方向 I_n 类梯度值 $k_n = 0.5$

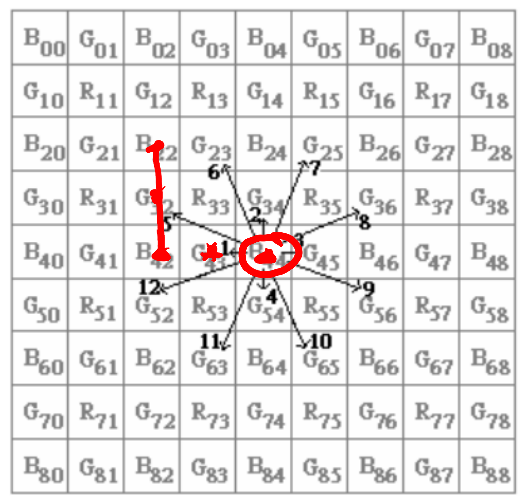
$$W_n(i,j) = \left(\frac{1}{1+I_n(i,j)}\right) \Big/ \sum\left(\frac{1}{1+I_n(i,j)}\right)$$

算法精讲-方向加权法

- RB缺失的G通道的插值

B上的G的插值.

R →



n	v_n	h_n	n	v_n	h_n
1	0	-1	2	-1	0
3	0	+1	4	+1	0
5	-1	-2	6	-2	-1
7	-2	+1	8	-1	+2
9	+1	+2	10	+2	+1
11	+2	-1	12	+1	-2

$W_n(i,j)$

$$G(i,j) = B(i,j) + \sum_{n=1}^{12} W_n(i,j) * K_{b,n}(i+v_n, j+h_n)$$

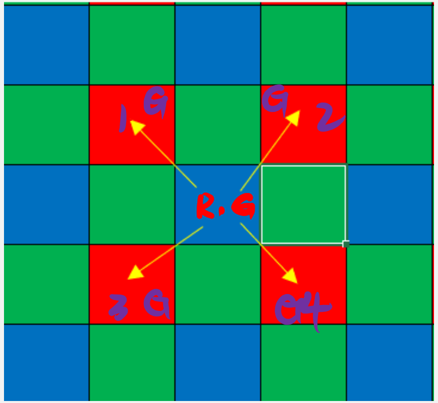
$G(i,j) - B(i,j) = \Delta$色差.

$$K_{b,n}(i+v_n, j+h_n) = G(i+v_n, j+h_n) - B(i+v_n, j+h_n)$$

左右的B 和 上下的B 的 mean

算法精讲-方向加权法

- RB缺失的BR的插值



B R => B (Row)

TABLE 2. POSITIONS OF NEARBY SAMPLES IN STEP 2.

n	v_n'	h_n'	n	v_n'	h_n'
1	-1	-1	2	-1	+1
3	+1	+1	4	+1	-1

$$I_n(i,j) = abs[P(i+v_n', j+h_n') - P(i-v_n', j-h_n')] + abs[P(i+2v_n', j+2h_n') - P(i,j)]$$

$$W_n(i,j) = \frac{1}{1+I_n(i,j)} \Big/ \sum_{n=1}^{4} \frac{1}{(1+I_n(i,j))}$$

$$R(i,j) = G(i,j) - \sum_{n=1}^{4} W_n(i,j) \times K_{rn}(i+v_n', j+h_n')$$

Δ

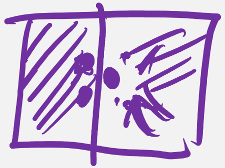
$$K_{rn}(i+v_n', j+h_n') = G(i+v_n', j+h_n') - R(i+v_n', j+h_n')$$

Δ

算法精讲-方向加权法

- G缺失的BR的插值

n	v_n	h_n	n	v_n	h_n
1	0	-1	2	-1	0
3	0	+1	4	+1	0
5	-1	-2	6	-2	-1
7	-2	+1	8	-1	+2
9	+1	+2	10	+2	+1
11	+2	-1	12	+1	-2



$$I_n(i,j) = \kappa_n(abs(P(i+v_n, j+h_n) - P(i-v_n, j-h_n)) + abs(P(i+2v_n, j+2h_n) - P(i,j)))$$

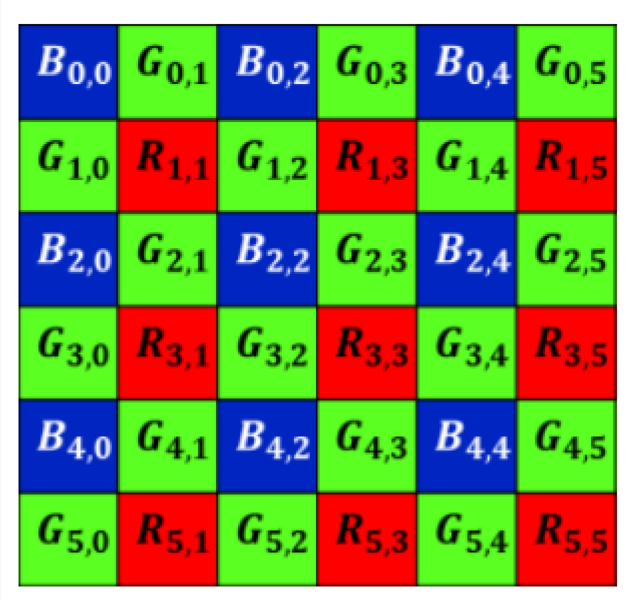
$$w_n(i,j) = \left(\frac{1}{1+I_n(i,j)}\right) \Big/ \sum_{n=1}^{12} \frac{1}{1+I_n(i,j)},$$

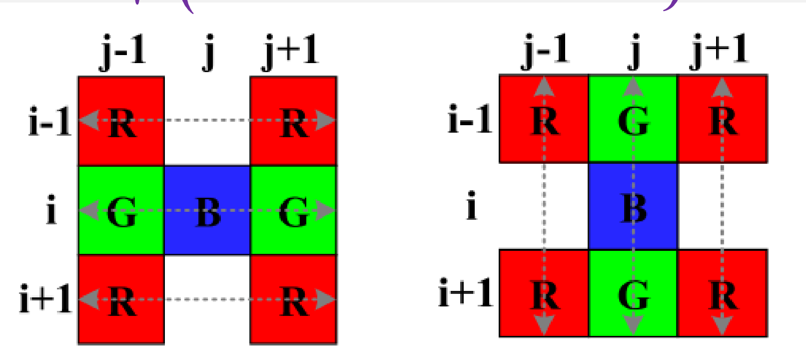
$$R(i,j) = G(i,j) - \sum_{n=1}^{12} w_n(i,j) * K_{r,n}(i+v_n, j+h_n),$$

算法精讲-基于边缘检测的方法

- 计算水平和竖直梯度

$$\begin{cases} e_h = \frac{1}{4}\left(\begin{aligned} &|C(i-1,j-1) - C(i-1,j+1)| + \\ &2|C(i-1,j) - C(i+1,j)| + \\ &|C(i+1,j-1) - C(i+1,j+1)| \end{aligned}\right) \\ e_v = \frac{1}{4}\left(\quad \right) \end{cases}$$

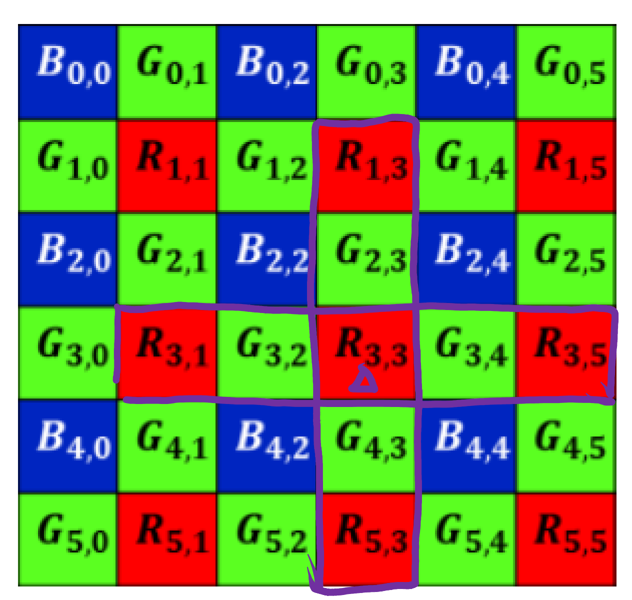


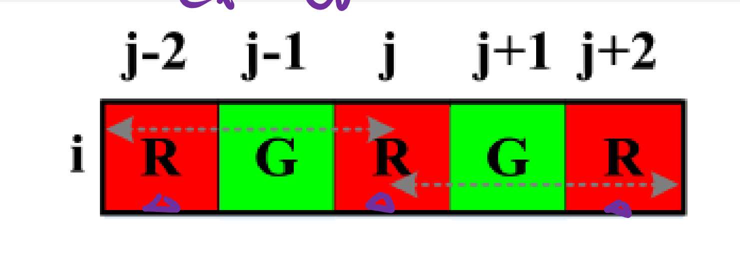


(a) (b)

- 计算缺失的G

$$W_h = \frac{\frac{1}{e_h}}{\frac{1}{e_h}+\frac{1}{e_v}} \qquad W_v = \frac{\frac{1}{e_v}}{\frac{1}{e_v}+\frac{1}{e_h}}$$





$$\begin{cases} \hat{G}_h(i,j) = \tilde{G}_h(i,j) + \Delta C_h(i,j) \\ \hat{G}_v(i,j) = \tilde{G}_v(i,j) + \Delta C_v(i,j) \end{cases}$$

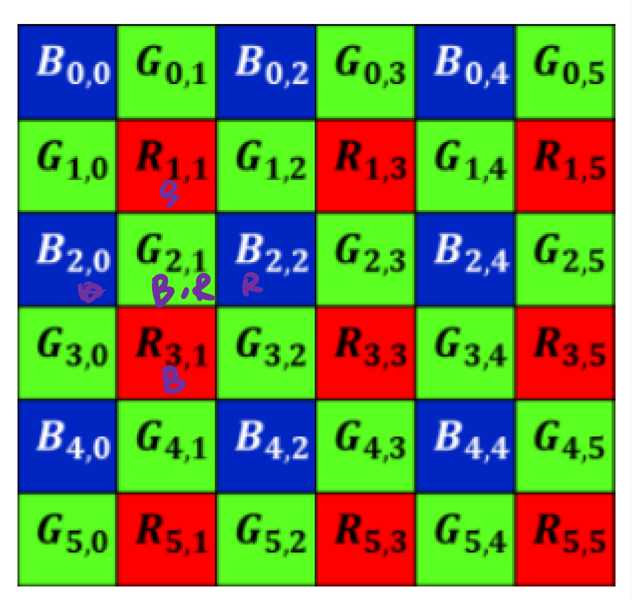
$$\Delta G_h(i,j) = \frac{1}{2}\left(\frac{1}{2}\left(C(i,j) - C(i,j-2)\right) + \frac{1}{2}\left(C(i,j) - C(i,j+2)\right)\right)$$

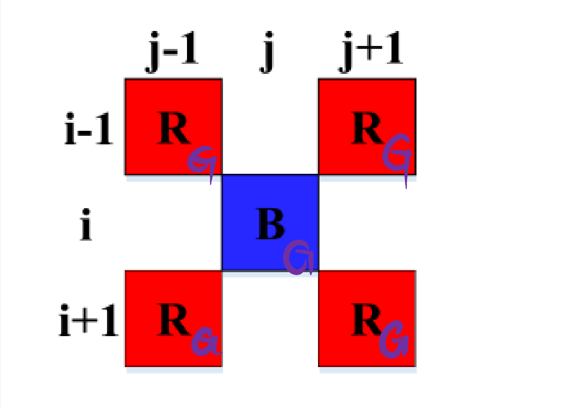
$$G'(i,j) = W_h(i,j) * \hat{G}_h(i,j) + W_v(i,j) * \hat{G}_v(i,j)$$

算法精讲-基于边缘检测的方法

- 计算缺失的RB

R => B　　　　B => R





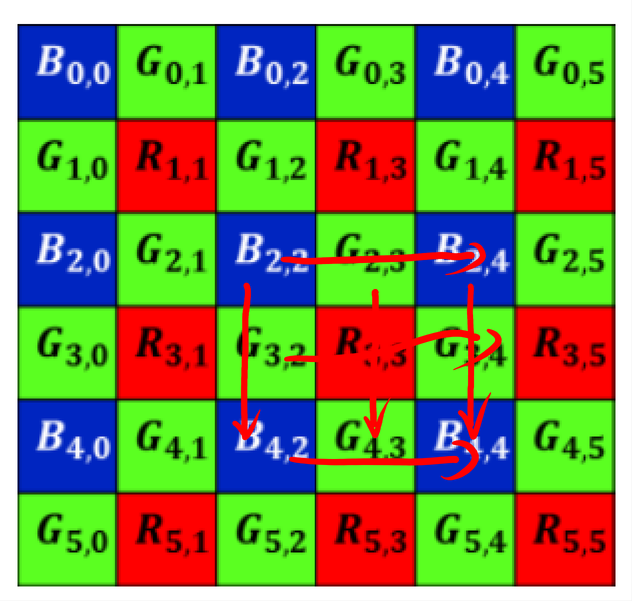
$\cancel{d_{GC} = \frac{1}{n} \sum G'(i+k, j+l) - C(i+k, j+l)}$

$R'(i,j) = G'(i,j) + d_{GC}$

G → BR

算法精讲-自适应边缘增强法

- 计算缺失的G



$$DV_{i,j} = |RB(i-1,j-1) - RB(i+1,j-1)| + |G(i-1,j) - G(i+1,j)| + |RB(i-1,j+1) - RB(i+1,j+1)|$$

$DH(i,j)$

$TD(i,j) = DV + DH$

算法精讲-自适应边缘增强法

- 计算缺失的G

$B_{0,0}$	$G_{0,1}$	$B_{0,2}$	$G_{0,3}$	$B_{0,4}$	$G_{0,5}$
$G_{1,0}$	$R_{1,1}$	$G_{1,2}$	$R_{1,3}$	$G_{1,4}$	$R_{1,5}$
$B_{2,0}$	$G_{2,1}$	$B_{2,2}$	$G_{2,3}$	$B_{2,4}$	$G_{2,5}$
$G_{3,0}$	$R_{3,1}$	$G_{3,2}$	$R_{3,3}$	$G_{3,4}$	$R_{3,5}$
$B_{4,0}$	$G_{4,1}$	$B_{4,2}$	$G_{4,3}$	$B_{4,4}$	$G_{4,5}$
$G_{5,0}$	$R_{5,1}$	$G_{5,2}$	$R_{5,3}$	$G_{5,4}$	$R_{5,5}$

$TD = DH + DV$

$TD < Th$ 那么平坦

$$G(i,j) = \frac{3}{8}\left(G(i,j-1) + G(i,j+1)\right) + \frac{1}{8}\left(G(i-1,j) + G(i+1,j)\right) + R - \frac{1}{2}\left(\frac{1}{2}\left(R + R(i,j-2)\right) + \frac{1}{2}\left(R + R(i,j+2)\right)\right)$$

$TD > Th,\ DH < DV$

$$G(i,j) = \frac{1}{2}\left(G(i,j-1) + G(i,j+1)\right) + (R - \bar{R})$$

$TD > Th,\ DH < DV$

$$G(i,j) = \frac{1}{8}\bar{G_h} + \frac{3}{8}\bar{G_v} + \frac{1}{2}(R - \bar{R})$$

算法精讲-自适应边缘增强法

- 计算缺失的RB

$B_{0,0}$	$G_{0,1}$	$B_{0,2}$	$G_{0,3}$	$B_{0,4}$	$G_{0,5}$
$G_{1,0}$	$R_{1,1}$	$G_{1,2}$	$R_{1,3}$	$G_{1,4}$	$R_{1,5}$
$B_{2,0}$	$G_{2,1}$	$B_{2,2}$	$G_{2,3}$	$B_{2,4}$	$G_{2,5}$
$G_{3,0}$	$R_{3,1}$	$G_{3,2}$	$R_{3,3}$ G	$G_{3,4}$	$R_{3,5}$
$B_{4,0}$	$G_{4,1}$	$B_{4,2}$	$G_{4,3}$	$B_{4,4}$	$G_{4,5}$
$G_{5,0}$	$R_{5,1}$	$G_{5,2}$	$R_{5,3}$	$G_{5,4}$	$R_{5,5}$

$TD < Th$ 平坦区

补偿

$$B(3,3) = \bar{B}_n + G(3,3) - \bar{G}_n$$

$DH < DV$

$$B(3,3) = \bar{B}_n + G(3,3) - \frac{1}{8}\left(3[G(i-1,j) + G(i+1,j)] + G(i,j-1) + G(i,j+1)\right)$$

$DH > DV$

$$\bar{B}_n + G(3,3) - \frac{1}{8}\left(G(i-1,j) + G(i+1,j) + 3(G(i,j+1) + G(i,j-1))\right)$$

算法精讲-自适应边缘增强法

- 计算缺失的RB

1 1 1 1
0 1 1 1

$B_{0,0}$	$G_{0,1}$	$B_{0,2}$	$G_{0,3}$	$B_{0,4}$	$G_{0,5}$
$G_{1,0}$	$R_{1,1}$	$G_{1,2}$	$R_{1,3}$	$G_{1,4}$	$R_{1,5}$
$B_{2,0}$	$G_{2,1}$	$B_{2,2}$	$G_{2,3}$ RB	$B_{2,4}$	$G_{2,5}$
$G_{3,0}$	$R_{3,1}$	$G_{3,2}$	$R_{3,3}$	$G_{3,4}$	$R_{3,5}$
$B_{4,0}$	$G_{4,1}$	$B_{4,2}$	$G_{4,3}$	$B_{4,4}$	$G_{4,5}$
$G_{5,0}$	$R_{5,1}$	$G_{5,2}$	$R_{5,3}$	$G_{5,4}$	$R_{5,5}$

$\frac{1}{2}$ $\frac{1}{4}$ $\frac{1}{8}$

$$RB(i,j)=\frac{1}{2}(RB(i-1,j)+RB(i+1,j))+\frac{1}{2}G(i,j)-\frac{1}{8}[(G(i-1,j-1)+G(i-1,j+1)+G(i+1,j-1)+G(i+1,j+1)]$$

$$\frac{1}{2}(RB(i,j+1)+RB(i,j-1))+G(i,j)-\frac{1}{2}(g(i,j+1)+g(i,j-1))$$

移位 2倍数 2^n

02 算法代码实现

食鱼者

202203

wtzhu13

https: //gitee.com/wtzhu13

猪猪爱吃鱼

wtzhu__13

食鱼者

扫一扫上面的二维码图案，加我微信

See You!