



(12) 发明专利申请

(10) 申请公布号 CN 104021526 A

(43) 申请公布日 2014. 09. 03

(21) 申请号 201410245710. 0

(22) 申请日 2014. 06. 05

(66) 本国优先权数据

201310594594. 9 2013. 11. 21 CN

(71) 申请人 上海海洋大学

地址 201306 上海市浦东新区沪城环路 999 号

(72) 发明人 袁小华 黄冬梅 王振华 常英立 王令群

(51) Int. Cl.

G06T 5/00 (2006. 01)

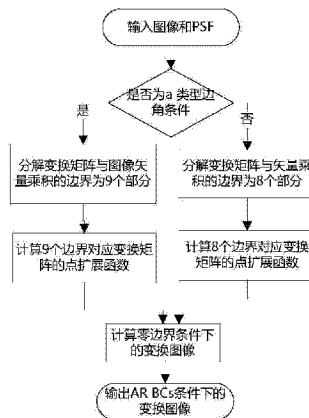
权利要求书6页 说明书11页 附图5页

(54) 发明名称

AR 边界条件下图像模糊矩阵与矢量乘积的替代计算方法

(57) 摘要

公开一种在 Anti-reflective 边界条件下, 大型模糊矩阵与图像矢量乘积 (乘积一), 以及模糊矩阵转置与图像矢量乘积 (乘积二) 的替代计算方法, 包括: 1) 按照边角是 a 类型还是 b 类型, 将乘积一和乘积二, 分别化为多个矩阵与图像矢量的乘积之和, 使分解结果中, 正好包含一个能对应 Zero 边界条件乘积的中心部分, 以及多个边界部分, 并且各个分解矩阵能带可利用的分块结构; 2) 构造各个分块矩阵的点扩展函数; 3) 用各分块矩阵的点扩展函数与图像, 或图像某个边界间的卷积, 代替计算乘积一和乘积二的各分解部分; 以及 4) 计算不同边角类型时的乘积一和乘积二。所公开的计算方法, 可应用到 Anti-reflective 边界条件下的大型图像滤波与图像恢复中, 用于解决其中的乘积一和乘积二难于计算的问题。



1. 一种 AR 边界条件下的图像恢复的替代计算方法,所述方法包括在 AR 边界条件下:

- (1) 依据边角类型,对乘积一和乘积二进行分解。
- (2) 依据边角类型,构造乘积一和乘积二分解公式中各矩阵的点扩展函数;
- (3) 依据边角类型,计算乘积一和乘积二分解公式中的各边界部分;
- (4) 计算乘积一和乘积二的中心部分;
- (5) 依据边角类型,计算 AR BCs 下的乘积一和乘积二。

2. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法,其特征在于,步骤

(1) 中所述的乘积一的分解方法,具体为:

a 边角条件下,分解为 $g_{B_Ar_a} = (K_{B_tt} - K_{B_th} + K_{B_tr} - K_{B_ht} - K_{B_hh} + K_{B_rt} + K_{B_lh} + K_{B_hl} - K_{B_ll} + K_{B_pp}) f$;

b 边角条件下,将 g 分解为 $g_{B_Ar_b} = (K_{B_tt} - K_{B_th} + K_{B_tr} - K_{B_ht} + K_{B_hh} - K_{B_hr} + K_{B_rt} - K_{B_rh} + K_{B_rr}) f$ 。

3. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法,其特征在于,步骤

(1) 中所述乘积二的分解方法,具体为:

a 边角条件下,分解为 $g_{D_Ar_a} = (K_{D_tt} - K_{D_th} + K_{D_tr} - K_{D_ht} - K_{D_hh} + K_{D_rt} + K_{D_lh} + K_{D_hl} - K_{D_ll} + K_{D_pp}) f$;

b 边角条件下,将 g 分解为 $g_{D_Ar_b} = (K_{D_tt} - K_{D_th} + K_{D_tr} - K_{D_ht} + K_{D_hh} - K_{D_hr} + K_{D_rt} - K_{D_rh} + K_{D_rr}) f$ 。

4. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法,其特征在于,步骤

(2) 中所述的乘积一的各分解矩阵的点扩展函数,在 a 边角类型时,具体构造方法为:

(1) 对 K_{B_th} , 构造 2 个点扩展函数 $k_{B_th_1} = \text{fliplr}(k(:, q_1+1:q))$, $k_{B_th_2} = \text{fliplr}(k(:, 1:q_1+1))$;

(2) K_{B_tr} 的 2 个大小为 $p \times (q_1+1)$ 的点扩展函数的元素,分别由 $k_{B_tr_1}(i,j) = k(i, q_1+j) + 2 \sum_{s_1=q_1+1+j}^q k(i, s_1)$, $1 \leq j \leq q_1$, $k_{B_tr_1}(i, q_1+1) = k(i, q_1+1)$ 和 $k_{B_tr_2}(i,j) = k(i, j) + 2 \sum_{s_1=1}^{j-1} k(i, s_1)$, $2 \leq j \leq q_1+1$, $k_{B_tr_2}(i, 1) = k(i, 1)$, $i = 1, \dots, p$ 计算;

(3) 对 K_{B_ht} , 构造 2 个点扩展函数 $k_{B_ht_1} = \text{flipud}(k(p_1+1:p, :))$, $k_{B_ht_2} = \text{flipud}(k(1:p_1+1, :))$;

(4) 对 K_{B_hh} , 构造 4 个点扩展函数 $k_{B_hh_1} = \text{flipud}(\text{fliplr}(k(p_1+1:p, q_1+1:q)))$, $k_{B_hh_2} = \text{flipud}(\text{fliplr}(k(p_1+1:p, 1:q_1+1)))$, $k_{B_hh_3} = \text{flipud}(\text{fliplr}(k(1:p_1+1, q_1+1:q)))$, $k_{B_hh_4} = \text{flipud}(\text{fliplr}(k(1:p_1+1, 1:q_1+1)))$;

(5) 对 K_{B_rt} , 构造 2 个点扩展函数 $k_{B_rt_1}(i,:) = k(p_1+2-i,:) + 2 \sum_{s_1=p_1+i+1}^p k(s_1,:)$, $1 \leq i \leq p_1$, $k_{B_rt_1}(p_1+1, :) = k(p, :)$ 和 $k_{B_rt_2}(i,:) = k(i,:) + 2 \sum_{s_1=1}^{i-1} k(s_1,:)$, $2 \leq i \leq p_1+1$, $k_{B_rt_2}(1, :) = k(1, :)$;

(6) K_{B_lh} 与 K_{B_hh} 的点扩展函数相同,即 $k_{B_lh_i} = k_{B_hh_i}$, $i = 1, \dots, 4$;

(7) K_{B_hl} 与 K_{B_hh} 的点扩展函数相同,即 $k_{B_hl_i} = k_{B_hh_i}$, $i = 1, \dots, 4$;

(8) K_{B_ll} 与 K_{B_hh} 的点扩展函数相同,即 $k_{B_ll_i} = k_{B_hh_i}$, $i = 1, \dots, 4$;

(9) K_{pp} 的 4 个点扩展函数 k_{pp_t} , $t = 1, \dots, 4$, 分别由公式

$$k_{pp_1}(i,j) = 2 \sum_{s_1=1}^{p_1+1-i} \sum_{s_2=1}^{q_1+1-j} k(s_1, s_2), \quad k_{pp_2}(i,j) = 2 \sum_{s_1=1}^{p_1+1-i} \sum_{s_2=q_1+1-j}^q k(s_1, s_2),$$

$$\mathbf{k}_{B_{pp_3}}(i,j)=2 \sum_{s_1=p_1+i+1}^p \sum_{s_2=1}^{q_1-j+i} \mathbf{k}(S_1,S_2), \text{ 和 } \mathbf{k}_{B_{pp_4}}(i,j)=2 \sum_{s_1=p_1+i+1}^p \sum_{s_2=q_1+1+j}^q \mathbf{k}(S_1,S_2), \quad 1 \leq i \leq p_1, 1 \leq j \leq q_1$$

构造。

5. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法,其特征在于,步骤 (2) 中所述乘积一的各个分解矩阵的点扩展函数,在 b 边角类型时,具体构造方法为:

(1) 模糊矩阵 $K_{B_{th}}$ 、 $K_{B_{tr}}$ 、 $K_{B_{ht}}$ 、 $K_{B_{hh}}$ 和 $K_{B_{rt}}$ 的点扩展函数,其构造方法与上述权利 4 中同名矩阵的点扩展函数的构造相同;

(2) $K_{B_{hr}}$ 的 4 个点扩展函数与 $K_{B_{tr}}$ 点扩展函数相关,具体为:

$$k_{B_{hr_1}} = \text{flipud}(k_{B_{tr_1}}(1:p_1+1, :)), k_{B_{hr_2}} = \text{flipud}(k_{B_{tr_2}}(p_1+1:\text{end}, :)), \\ k_{B_{hr_3}} = \text{flipud}(k_{B_{tr_1}}(\text{end}-p_1:\text{end}, :)), k_{B_{hr_4}} = \text{flipud}(k_{B_{tr_2}}(1:p_1+1, :));$$

(3) $K_{B_{rh}}$ 的 4 个点扩展函数与 $K_{B_{rt}}$ 的点扩展函数相关,具体为:

$$k_{B_{rh_1}} = \text{fliplr}(\text{flipud}(k_{B_{rt_1}}(1:p_1+1, q_1+1:\text{end}))), \\ k_{B_{rh_2}} = \text{fliplr}(\text{flipud}(k_{B_{rt_1}}(1:p_1+1, 1:q_1+1))), k_{B_{rh_3}} = \text{flipud}(k_{B_{rt_2}}(1:p_1+1, \text{end}-q_1:\text{end})), \\ k_{B_{rh_4}} = \text{flipud}(k_{B_{rt_2}}(1:p_1+1, 1:q_1+1));$$

(4) $K_{B_{rr}}$ 的 4 点扩展函数与 $k_{B_{tr_1}}$ 和 $k_{B_{tr_2}}$ 相关,计算方法分别为:

$$\mathbf{k}_{B_{rr_1}}(i,:) = \mathbf{k}_{B_{tr_1}}(p_1+i,:) + 2 \sum_{s_1=p_1+i+1}^p \mathbf{k}_{B_{tr_1}}(S_1,:), \quad 1 \leq i \leq p_1, \quad k_{B_{rr_1}}(p_1+1, :) = k_{B_{tr_1}}(1, :);$$

$$\mathbf{k}_{B_{rr_2}}(i,:) = \mathbf{k}_{B_{tr_2}}(p_1+i,:) + 2 \sum_{s_1=p_1+i+1}^p \mathbf{k}_{B_{tr_2}}(S_1,:), \quad 1 \leq i \leq p_1, \quad k_{B_{rr_2}}(p_1+1, :) = k_{B_{tr_2}}(p, :);$$

$$\mathbf{k}_{B_{rr_3}}(i,:) = \mathbf{k}_{B_{tr_1}}(i,:) + 2 \sum_{s_1=1}^{i-1} \mathbf{k}_{B_{tr_1}}(S_1,:), \quad 2 \leq i \leq p_1+1, \quad k_{B_{rr_3}}(1, :) = k_{B_{tr_1}}(p, :);$$

$$\mathbf{k}_{B_{rr_4}}(i,:) = \mathbf{k}_{B_{tr_2}}(i,:) + 2 \sum_{s_1=1}^{i-1} \mathbf{k}_{B_{tr_2}}(S_1,:), \quad 2 \leq i \leq p_1+1, \quad k_{B_{rr_4}}(1, :) = k_{B_{tr_2}}(1, :).$$

6. 根据权利要求 1 所述的 AR 边界条件下的图像去卷积方法,其特征在于,步骤 (2) 中所述的乘积二的各分解矩阵的点扩展函数,在 a 边角类型时,具体构造方法为:

(1) $K_{D_{th}}$ 的点扩展函数与实例中 $K_{B_{th}}$ 的点扩展函数相关,具体为:

$$k_{D_{th_i}} = \text{flipud}(k_{B_{th_i}}), \quad i = 1, 2;$$

(2) $K_{D_{tr}}$ 的点扩展函数与实例中 $K_{B_{tr}}$ 的点扩展函数相关,具体为:

$$k_{D_{tr_i}} = \text{flipud}(k_{B_{tr_i}}), \quad i = 1, 2;$$

(3) $K_{D_{ht}}$ 的点扩展函数与实例中 $K_{B_{ht}}$ 的点扩展函数相关,具体为:

$$k_{D_{ht_i}} = \text{fliplr}(k_{B_{ht_i}}), \quad i = 1, 2;$$

(4) $K_{D_{hh}}$ 的点扩展函数与 $K_{B_{hh}}$ 的相同,即 $k_{D_{hh_i}} = k_{B_{hh_i}}, i = 1, \dots, 4;$

(5) 对 $K_{D_{rt}}$ 的点扩展函数与 $K_{B_{rt}}$ 的相关,具体为:的构造 2 个点扩展函数

$$k_{D_{rt_i}} = \text{fliplr}(\text{flipud}(k_{B_{rt_i}})), \quad i = 1, 2;$$

(6) $K_{D_{lh}}$ 与 $K_{B_{hh}}$ 的点扩展函数相同,即 $k_{D_{lh_i}} = k_{B_{hh_i}}, i = 1, \dots, 4;$

(7) $K_{D_{hl}}$ 与 $K_{B_{hl}}$ 的点扩展函数相同,即 $k_{D_{hl_i}} = k_{B_{hl_i}}, i = 1, \dots, 4;$

(8) $K_{D_{ll}}$ 与 $K_{B_{ll}}$ 的点扩展函数相同,即 $k_{D_{ll_i}} = k_{B_{ll_i}}, i = 1, \dots, 4;$

(9) $K_{D_{pp}}$ 的点扩展函数与 $K_{B_{pp}}$ 的相同,即 $k_{D_{pp_i}} = k_{B_{pp_i}}, i = 1, \dots, 4.$

7. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法,其特征在于,步骤 (2) 中所述乘积二的各个分解矩阵的点扩展函数,在 b 边角类型时,具体构造方法为:

(1) K_{D_th} 、 K_{D_tr} 、 K_{D_ht} 、 K_{D_hh} 、 K_{D_rt} 的点扩展函数构造方法,与上相同;

(2) K_{D_hr} 的 4 个点扩展函数与 K_{B_hr} 点扩展函数相关,具体为:

$$k_{D_hr_i} = \text{fliplr}(k_{B_hr_i}), i = 1, \dots, 4;$$

(3) K_{D_rh} 的 4 个点扩展函数与 K_{B_rh} 的点扩展函数相同,具体为:

$$k_{D_rh_i} = k_{B_rh_i}, i = 1, \dots, 4;$$

(4) K_{D_rr} 的点扩展函数与 K_{B_rr} 的相关,即: $k_{D_rr_i} = \text{fliplr}(\text{flipud}(k_{B_rr_i})), i = 1, \dots, 4$ 。

8. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法,其特征在于,在当边角类型为 a 时,步骤 (3) 中所述的乘积一分解公式中各个边界部分所对应的图像矩阵 G_{B_th} 、 G_{B_tr} 、 G_{B_rt} 、 G_{B_ht} 、 G_{B_hh} 、 G_{B_hl} 、 G_{B_lh} 、 G_{B_ll} 和 G_{B_pp} ,具体的计算方法为:

(1) 对 G_{B_th} , 计算 $r_{B_th_1} = \text{conv}(k_{B_th_1}, F(:, 1:q_1+1))$, $r_{B_th_2} = \text{conv}(k_{B_th_2}, F(:, n-q_1:n))$, 则 $G_{B_th}(:, 1:q_1+1) = \text{fliplr}(r_{B_th_1}(p_1+1:\text{end}-p_1, 1:q_1+1))$, $G_{B_th}(:, n-q_1:n) = \text{fliplr}(r_{B_th_2}(p_1+1:\text{end}-p_1, q_1+1:\text{end}))$, G_{B_th} 的其他元素为 0;

(2) 对 G_{B_tr} , 计算 $r_{B_tr_1} = \text{conv2}(\text{fliplr}(\text{flipud}(k_{B_tr_1})), F(:, 1))$, $r_{B_tr_2} = \text{conv2}(\text{fliplr}(k_{B_tr_2}), F(:, n))$, $G_{B_tr}(:, 1:q_1+1) = \text{fliplr}(r_{B_tr_1}(p_1+1:\text{end}-p_1, :))$, $G_{B_tr}(:, n-q_1:n) = \text{fliplr}(r_{B_tr_2}(p_1+1:\text{end}-p_1, :))$, G_{B_tr} 的其他元素为 0;

(3) 对 G_{B_ht} , 计算 $r_{B_ht_1} = \text{conv2}(k_{B_ht_1}, F(1:p_1+1, :))$, $r_{B_ht_2} = \text{conv2}(k_{B_ht_2}, F(m-p_1:m, :))$, 则 $G_{B_ht}(1:p_1+1, :) = \text{flipud}(r_{B_ht_1}(1:p_1+1, q_1+1:\text{end}-q_1))$, $G_{B_ht}(m-p_1:m, :) = \text{flipud}(r_{B_ht_2}(p_1+1:\text{end}, q_1+1:\text{end}-q_1))$, G_{B_ht} 的其他元素为 0;

(4) 对 G_{B_hh} , 计算 $r_{B_hh_1} = \text{conv2}(k_{B_hh_1}, F(1:p_1+1, 1:q_1+1))$, $r_{B_hh_2} = \text{conv2}(k_{B_hh_2}, F(1:p_1+1, n-q_1:n))$, $r_{B_hh_3} = \text{conv2}(k_{B_hh_3}, F(m-p_1:m, 1:q_1+1))$, $r_{B_hh_4} = \text{conv2}(k_{B_hh_4}, F(m-p_1:m, n-q_1:n))$, 则 $G_{B_hh}(1:p_1+1, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_hh_1}(1:p_1+1, 1:q_1+1)))$, $G_{B_hh}(1:p_1+1, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_hh_2}(1:p_1+1, \text{end}-q_1:\text{end})))$, $G_{B_hh}(m-p_1:m, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_hh_3}(\text{end}-p_1:\text{end}, 1:q_1+1)))$, $G_{B_hh}(m-p_1:m, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_hh_4}(\text{end}-p_1:\text{end}, \text{end}-q_1:\text{end})))$, G_{B_hh} 的其他元素为 0;

(5) 对 G_{B_rt} , 计算 $r_{B_rt_1} = \text{conv2}(k_{B_rt_1}(1:p_1+1, :), F(1, :))$, $r_{B_rt_2} = \text{conv2}(\text{flipud}(k_{B_rt_2}(1:p_1+1)), F(m, :))$, 则 $G_{B_rt}(1:p_1+1, :) = r_{B_rt_1}(:, q_1+1:\text{end}-q_1)$, $G_{B_rt}(m-p_1:m, :) = r_{B_rt_2}(:, q_1+1:\text{end}-q_1)$, G_{B_rt} 的其他元素为 0;

(6) 对 G_{B_lh} , 计算 $r_{B_lh_1} = \text{conv2}(k_{B_lh_1}, F(1, 1:q_1+1))$, $r_{B_lh_2} = \text{conv2}(k_{B_lh_2}, F(1, n-q_1:n))$, $r_{B_lh_3} = \text{conv2}(k_{B_lh_3}, F(m, 1:q_1+1))$, $r_{B_lh_4} = \text{conv2}(k_{B_lh_4}, F(m, n-q_1:n))$, 则 $G_{B_lh}(1:p_1+1, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_lh_1}(1:p_1+1, 1:q_1+1)))$, $G_{B_lh}(1:p_1+1, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_lh_2}(1:p_1+1, \text{end}-q_1:\text{end})))$, $G_{B_lh}(m-p_1:m, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_lh_3}(\text{end}-p_1:\text{end}, 1:q_1+1)))$, $G_{B_lh}(m-p_1:m, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_lh_4}(1:\text{end}, q_1+1:\text{end})))$, G_{B_lh} 的其他元素为 0;

(7) 对于 G_{B_hl} , 计算 $r_{B_hl_1} = \text{conv2}(k_{B_hl_1}, F(1:p_1+1, 1))$, $r_{B_hl_2} = \text{conv2}(k_{B_hl_2}, F(1:p_1+1, n))$, $r_{B_hl_3} = \text{conv2}(k_{B_hl_3}, F(m-p_1:m, 1))$, $r_{B_hl_4} = \text{conv2}(k_{B_hl_4}, F(m-p_1:m, n))$, $G_{B_hl}(1:p_1+1, 1) = r_{B_hl_1}(p_1+1:m, 1)$, $G_{B_hl}(1:p_1+1, n) = r_{B_hl_2}(p_1+1:m, n)$, $G_{B_hl}(m-p_1:m, 1) = r_{B_hl_3}(p_1+1:m, 1)$, $G_{B_hl}(m-p_1:m, n) = r_{B_hl_4}(p_1+1:m, n)$, G_{B_hl} 的其他元素为 0;

$_{hl_4}$, $F(m-p_1:m, n)$), 则 $G_{B_{hl}}(1:p_1+1, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{hl_1}}(1:p_1+1, 1:q_1+1)))$, $G_{B_{hl}}(1:p_1+1, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{hl_2}}(1:p_1+1, \text{end}-q_1:\text{end})))$, $G_{B_{hl}}(m-p_1:m, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{hl_3}}(\text{end}-p_1:\text{end}, \text{end}-q_1:\text{end})))$, $G_{B_{hl}}(m-p_1:m, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{hl_4}}(\text{end}-p_1:\text{end}, :)))$, $G_{B_{hl}}$ 的其他元素为 0;

(8) 对 $G_{B_{ll}}$, 计算 $r_{B_{ll_1}} = \text{conv2}(k_{ll_1}, F(1, 1))$, $r_{B_{ll_2}} = \text{conv2}(k_{B_{ll_2}}, F(1, n))$, $r_{B_{ll_3}} = \text{conv2}(k_{B_{ll_3}}, F(m, 1))$, $r_{B_{ll_4}} = \text{conv2}(k_{B_{ll_4}}, F(m, n))$, 则 $G_{B_{ll}}(1:p_1+1, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{ll_1}}(1:p_1+1, 1:q_1+1)))$, $G_{B_{ll}}(1:p_1+1, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{ll_2}}(1:p_1+1, \text{end}-q_1:\text{end})))$, $G_{B_{ll}}(m-p_1:m, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{ll_3}}))$, $G_{B_{lh}}(m-p_1:m, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{lh_4}}))$, $G_{B_{ll}}$ 的其他元素为 0;

(9) 对 $G_{B_{pp}}$, 计算 $r_{B_{pp_1}} = \text{conv2}(\text{fliplr}(\text{flipud}(k_{B_{pp_1}})), F(m, n))$, $r_{B_{pp_2}} = \text{conv2}(\text{fliplr}(\text{flipud}(k_{B_{pp_2}})), F(m, 1))$, $r_{B_{pp_3}} = \text{conv2}(\text{fliplr}(k_{B_{pp_3}}), F(1, n))$, $r_{B_{pp_4}} = \text{conv2}(k_{B_{pp_4}}, F(1, 1))$, 则 $G_{B_{pp}}(m-p_1+1:m, n-q_1+1:n) = r_{B_{pp_1}}$, $G_{B_{pp}}(m-p_1+1:m, 1:q_1) = \text{fliplr}(r_{B_{pp_2}})$, $G_{B_{pp}}(1:p_1, n-q_1+1:n) = r_{B_{pp_3}}$, $G_{B_{pp}}(1:p_1, 1:q_1) = r_{B_{pp_4}}$, $G_{B_{pp}}$ 的其他元素为 0。

9. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法, 其特征在于, 在当边角类型为 b 时, 步骤 (3) 中所述的乘积一分解公式中各个边界部分所对应的图像矩阵, 其具体的计算方法为:

(1) $G_{B_{th}}$ 、 $G_{B_{tr}}$ 、 $G_{B_{rt}}$ 、 $G_{B_{ht}}$ 和 $G_{B_{hh}}$ 等的计算方法分别与第 8 实施例同名矩阵的计算方法相同;

(2) 对 $G_{B_{hr}}$, 计算 $r_{B_{hr_1}} = \text{conv2}(k_{B_{hr_1}}, F(1:p_1+1, 1))$, $r_{B_{hr_2}} = \text{conv2}(k_{B_{hr_2}}, F(1:p_1+1, n))$, $r_{B_{hr_3}} = \text{conv2}(k_{B_{hr_3}}, F(m-p_1:m, 1))$, $r_{B_{hr_4}} = \text{conv2}(k_{B_{hr_4}}, F(m-p_1:m, n))$, $G_{B_{hr}}(1:p_1+1, 1:q_1+1) = \text{flipud}(r_{B_{hr_1}}(1:p_1+1, :))$, $G_{B_{hr}}(1:p_1+1, n-q_1:n) = \text{flipud}(r_{B_{hr_2}}(1:p_1+1, :))$, $G_{B_{hr}}(m-p_1:m, 1:q_1+1) = \text{flipud}(r_{B_{hr_3}}(\text{end}-p_1:\text{end}, :))$, $G_{B_{hr}}(m-p_1:m, n-q_1:n) = \text{flipud}(r_{B_{hr_4}}(\text{end}-p_1:\text{end}, :))$, $G_{B_{hr}}$ 的其他元素为 0;

(3) 对 $G_{B_{rh}}$, 计算 $r_{B_{rh_1}} = \text{conv2}(k_{B_{rh_1}}, F(1, 1:q_1+1))$, $r_{B_{rh_2}} = \text{conv2}(k_{B_{rh_2}}, F(1, n-q_1:n))$, $r_{B_{rh_3}} = \text{conv2}(k_{B_{rh_3}}, F(m, 1:q_1+1))$, $r_{B_{rh_4}} = \text{conv2}(k_{B_{rh_4}}, F(m, n-q_1:n))$, 则 $G_{B_{rh}}(1:p_1+1, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{rh_1}}(:, 1:q_1+1)))$, $G_{B_{rh}}(1:p_1+1, n-q_1+1:n) = \text{fliplr}(\text{flipud}(r_{B_{rh_2}}(:, \text{end}-q_1:\text{end})))$, $G_{B_{rh}}(m-q_1:m, 1:q_1+1) = \text{fliplr}(r_{B_{rh_3}}(:, 1:q_1+1))$, $G_{B_{rh}}(m-p_1:m, n-q_1:n) = \text{fliplr}(r_{B_{rh_4}}(:, \text{end}-q_1:\text{end}))$, $G_{B_{rh}}$ 的其他元素为 0;

(4) 对 $G_{B_{rr}}$, 计算 $r_{B_{rr_1}} = \text{conv2}(\text{fliplr}(k_{B_{rr_1}}), F(1, 1))$, $r_{B_{rr_2}} = \text{conv2}(\text{fliplr}(k_{B_{rr_2}}), F(1, n))$, $r_{B_{rr_3}} = \text{conv2}(k_{B_{rr_3}}, F(m, 1))$, $r_{B_{rr_4}} = \text{conv2}(k_{B_{rr_4}}, F(m, n))$ 则 $G_{B_{rr}}(1:p_1+1, 1:q_1+1) = \text{fliplr}(r_{B_{rr_1}})$, $G_{B_{rr}}(1:p_1+1, n-q_1+1:n) = \text{fliplr}(r_{B_{rr_2}})$, $G_{B_{rr}}(m-p_1+1:m, 1:q_1+1) = r_{B_{rr_3}}$, $G_{B_{rr}}(m-p_1+1:m, n-q_1+1:n) = r_{B_{rr_4}}$, $G_{B_{rr}}$ 的其他元素为 0。

10. 根据权利要求 1 所述的 AR 边界条件下的图像恢复迭代方法, 其特征在于, 在步骤 (3) 中所述乘积二中的各个边界部分, 当边角类型为 a 时, 具体的计算方法为:

(1) 对 $G_{D_{th}}$, 计算 $r_{D_{th_1}} = \text{conv}(k_{D_{th_1}}, F(:, 1:q_1+1))$, $r_{D_{th_2}} = \text{conv}(k_{D_{th_2}}, F(:, n-q_1:n))$, 则 $G_{D_{th}}(:, 1:q_1+1) = \text{fliplr}(r_{D_{th_1}}(p_1+1:\text{end}-p_1, 1:q_1+1))$, $G_{D_{th}}(:, n-q_1:m) = \text{fliplr}(r_{D_{th_2}}(p_1+1:\text{end}-p_1, q_1+1:\text{end}))$, $G_{D_{th}}$ 的其他元素为 0;

(2) 对 $G_{D_{tr}}$, 计算 $r_{D_{tr}1} = \text{conv2}(k_{D_{tr}1}, F(:, 1:q_1+1))$, $r_{D_{tr}2} = \text{conv2}(k_{D_{tr}2}, F(:, n-q_1:n))$, $G_{D_{tr}}(:, 1) = r_{D_{tr}1}(p_1+1:\text{end}-p_1, q_1+1)$, $G_{D_{tr}}(:, n) = r_{D_{tr}2}(p_1+1:\text{end}-p_1, q_1+1)$, $G_{D_{tr}}$ 的其他元素为 0;

(3) 对 $G_{D_{ht}}$, 计算 $r_{D_{ht}1} = \text{conv2}(k_{D_{ht}1}, F(1:p_1+1, :))$, $r_{D_{ht}2} = \text{conv2}(k_{D_{ht}2}, F(m-p_1:m, :))$, 则 $G_{D_{ht}}(1:p_1+1, :) = \text{flipud}(r_{D_{ht}1}(1:p_1+1, q_1+1:\text{end}-q_1))$, $G_{D_{ht}}(m-p_1:m, :) = \text{flipud}(r_{D_{ht}2}(p_1+1:\text{end}, q_1+1:\text{end}-q_1))$, $G_{D_{ht}}$ 的其他元素为 0;

(4) 对 $G_{D_{hh}}$ 有 $G_{D_{hh}} = G_{hh}$;

(5) 对 $G_{D_{rt}}$, 计算 $r_{D_{rt}1} = \text{conv2}(k_{D_{rt}1}, F(1:p_1+1, :))$, $r_{D_{rt}2} = \text{conv2}(k_{D_{rt}2}(1:p_1+1), F(m-p_1:m, :))$, 则 $G_{D_{rt}}(1, :) = r_{D_{rt}1}(p_1+1, q_1+1:\text{end}-q_1)$, $G_{D_{rt}}(M, :) = \text{flipud}(r_{D_{rt}2}(p_1+1, q_1+1:\text{end}-q_1))$, $G_{D_{rt}}$ 的其他元素为 0;

(6) 对 $G_{D_{lh}}$, 计算 $r_{D_{lh}1} = \text{conv2}(k_{D_{lh}1}, F(1:p_1+1, 1:q_1+1))$, $r_{D_{lh}2} = \text{conv2}(k_{D_{lh}2}, F(1:p_1+1, n-q_1:n))$, $r_{D_{lh}3} = \text{conv2}(k_{D_{lh}3}, F(m-p_1:m, 1:q_1+1))$, $r_{D_{lh}4} = \text{conv2}(k_{D_{lh}4}, F(m-p_1:m, n-q_1:n))$, 则 $G_{D_{lh}}(1:, 1:q_1+1) = \text{fliplr}(r_{D_{lh}1}(p_1+1, 1:q_1+1))$, $G_{D_{lh}}(1:, n-q_1:n) = \text{fliplr}(r_{D_{lh}2}(p_1+1, \text{end}-q_1:\text{end}))$, $G_{D_{lh}}(m, 1:q_1+1) = \text{fliplr}(r_{D_{lh}3}(p_1+1, 1:q_1+1))$, $G_{D_{lh}}(m, n-q_1:n) = \text{fliplr}(r_{D_{lh}4}(p_1+1, \text{end}-q_1:\text{end}))$, $G_{D_{lh}}$ 的其他元素为 0;

(7) 对于 $G_{D_{hl}}$, 计算 $r_{D_{hl}1} = \text{conv2}(k_{D_{hl}1}, F(1:p_1+1, 1:q_1+1))$, $r_{D_{hl}2} = \text{conv2}(k_{D_{hl}2}, F(1:p_1+1, n-q_1:n))$, $r_{D_{hl}3} = \text{conv2}(k_{D_{hl}3}, F(m-p_1:m, 1:q_1+1))$, $r_{D_{hl}4} = \text{conv2}(k_{D_{hl}4}, F(m-p_1:m, n-q_1:n))$, 则 $G_{D_{hl}}(1:p_1+1, 1) = \text{flipud}(r_{D_{hl}1}(1:p_1+1, q_1+1))$, $G_{D_{hl}}(1:p_1+1, n) = \text{flipud}(r_{D_{hl}2}(1:p_1+1, q_1+1))$, $G_{D_{hl}}(m-p_1:m, 1) = \text{flipud}(r_{D_{hl}3}(\text{end}-p_1:\text{end}, q_1+1))$, $G_{D_{hl}}(m-p_1:m, n) = \text{flipud}(r_{D_{hl}4}(\text{end}-p_1:\text{end}, q_1+1))$, $G_{D_{hl}}$ 的其他元素为 0;

(8) 对 $G_{D_{ll}}$, 计算 $r_{D_{ll}1} = \text{conv2}(k_{D_{ll}1}, F(1:p_1, 1:q_1))$, $r_{D_{ll}2} = \text{conv2}(k_{D_{ll}2}, F(1:p_1+1, n-q_1:n))$, $r_{D_{ll}3} = \text{conv2}(k_{D_{ll}3}, F(m-p_1:m, 1:q_1+1))$, $r_{D_{ll}4} = \text{conv2}(k_{D_{ll}4}, F(m-p_1:m, n-q_1:n))$, 则 $G_{D_{ll}}(1) = r_{D_{ll}1}(p_1+1, q_1+1)$, $G_{D_{ll}}(1:p_1+1, n) = r_{D_{ll}2}(p_1+1, q_1+1)$, $G_{D_{ll}}(m, 1) = r_{D_{ll}3}(p_1+1, q_1+1)$, $G_{D_{ll}}(m, n) = r_{D_{ll}4}(p_1+1, q_1+1)$, $G_{D_{ll}}$ 的其他元素为 0;

(9) 对 $G_{D_{pp}}$, 计算 $r_{D_{pp}1} = \text{conv2}(\text{fliplr}(\text{flipud}(k_{D_{pp}1})), F(1:p_1, 1:q_1))$, $r_{D_{pp}2} = \text{conv2}(\text{flipud}(k_{D_{pp}2}), F(1:p_1, n-q_1+1:n))$, $r_{D_{pp}3} = \text{conv2}(\text{fliplr}(k_{D_{pp}3}), F(m-p_1+1:m, 1:q_1))$, $r_{D_{pp}4} = \text{conv2}(k_{D_{pp}4}, F(1, 1))$, 则 $G_{D_{pp}}(1, 1) = r_{D_{pp}1}(p_1, q_1)$, $G_{D_{pp}}(1, n) = r_{D_{pp}2}(p_1, q_1)$, $G_{D_{pp}}(m, 1) = r_{D_{pp}3}(p_1, q_1)$, $G_{D_{pp}}(m, n) = r_{D_{pp}4}(p_1, q_1)$, $G_{D_{pp}}$ 的其他元素为 0。

11. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法, 其特征在于, 在步骤 (3) 中所述的乘积二中的各个边界部分, 当边角类型为 b 时, 具体的计算方法为:

(1) $G_{D_{th}}$ 、 $G_{D_{tr}}$ 、 $G_{D_{rt}}$ 、 $G_{D_{ht}}$ 、 $G_{D_{hh}}$ 计算方法与第 10 实施例同名矩阵的计算方法相同;

(2) 对 $G_{D_{hr}}$, 计算 $r_{D_{hr}1} = \text{conv2}(k_{D_{hr}1}, F(1:p_1+1, 1:q_1+1))$, $r_{D_{hr}2} = \text{conv2}(k_{D_{hr}2}, F(1:p_1+1, n-q_1:n))$, $r_{D_{hr}3} = \text{conv2}(k_{D_{hr}3}, F(m-p_1:m, 1:q_1+1))$, $r_{D_{hr}4} = \text{conv2}(k_{D_{hr}4}, F(m-p_1:m, n-q_1:n))$, $G_{D_{hr}}(1:p_1+1, 1) = \text{flipud}(r_{D_{hr}1}(1:p_1+1, q_1+1))$, $G_{D_{hr}}(1:p_1+1, n) = \text{flipud}(r_{D_{hr}2}(1:p_1+1, q_1+1))$, $G_{D_{hr}}(m-p_1:m, 1) = \text{flipud}(r_{D_{hr}3}(\text{end}-p_1:\text{end}, q_1+1))$, $G_{D_{hr}}$

$r_{hr}(m-p_1:m, n) = \text{flipud}(r_{D_{hr_4}}(\text{end}-p_1:\text{end}, q_1+1))$, $G_{D_{hr}}$ 的其他元素为 0;

(3) 对 $G_{D_{rh}}$, 计算 $r_{D_{rh_1}} = \text{conv2}(k_{D_{rh_1}}, F(1:p_1+1, 1:q_1+1))$, $r_{D_{rh_2}} = \text{conv2}(k_{D_{rh_2}}, F(1:p_1+1, n-q_1:n))$, $r_{D_{rh_3}} = \text{conv2}(k_{D_{rh_3}}, F(m-p_1:m, 1:q_1+1))$, $r_{D_{rh_4}} = \text{conv2}(k_{D_{rh_4}}, F(m-p_1:m, n-q_1:n))$, 则 $G_{D_{rh}}(1, 1:q_1+1) = \text{fliplr}(r_{D_{rh_1}}(p_1+1, 1:q_1+1))$, $G_{D_{rh}}(1, n-q_1+1:n) = \text{fliplr}(r_{D_{rh_2}}(p_1+1, \text{end}-q_1:\text{end}))$, $G_{D_{rh}}(m, 1:q_1+1) = \text{fliplr}(r_{D_{rh_3}}(p_1+1, 1:q_1+1))$, $G_{D_{rh}}(m, n-q_1:n) = \text{fliplr}(r_{D_{rh_4}}(p_1+1, \text{end}-q_1:\text{end}))$, $G_{D_{rh}}$ 的其他元素为 0;

(4) 对 $G_{D_{rr}}$, 计算 $r_{D_{rr_1}} = \text{conv2}(k_{D_{rr_1}}, F(1:p_1+1, 1:q_1+1))$, $r_{D_{rr_2}} = \text{conv2}(k_{D_{rr_2}}, F(1:p_1+1, n-q_1:n))$, $r_{D_{rr_3}} = \text{conv2}(k_{D_{rr_3}}, F(m-p_1:m, 1:q_1+1))$, $r_{D_{rr_4}} = \text{conv2}(k_{D_{rr_4}}, F(m-p_1:m, n-q_1:n))$, 则 $G_{D_{rr}}(1, 1) = r_{D_{rr_1}}(p_1+1, q_1+1)$, $G_{D_{rr}}(1, n) = r_{D_{rr_2}}(p_1+1, q_1+1)$, $G_{D_{rr}}(m, 1) = r_{D_{rr_3}}(p_1+1, q_1+1)$, $G_{D_{rr}}(m, n) = r_{D_{rr_4}}(p_1+1, q_1+1)$, $G_{D_{rr}}$ 的其他元素为 0。

12. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法, 其特征在于, 在步骤 (4) 中所述的 Zero BCs 下乘积一, 计算方法为 $G_{B_{zero}} = \text{conv2}(k, f)$ 。

13. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法, 其特征在于, 在步骤 (4) 中所述的 Zero BCs 下乘积二, 计算方法为 $G_{D_{zero}} = \text{conv2}(k', f)$ 。

14. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法, 其特征在于, 在步骤 (5) 中所述的乘积一的计算方法, 当边角类型为 a 时, 为 $G_{B_{Ar_a}} = G_{B_{Zero}} - G_{B_{th}} + G_{B_{tr}} + G_{B_{rt}} - G_{B_{ht}} - G_{B_{hh}} + G_{B_{hl}} + G_{B_{lh}} - G_{B_{ll}} + G_{B_{pp}}$ 。

15. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法, 其特征在于, 在步骤 (5) 中所述的乘积一的计算方法, 当边角类型为 b 时, 为 $G_{B_{Ar_b}} = G_{B_{Zero}} - G_{B_{th}} + G_{B_{tr}} - G_{B_{ht}} + G_{B_{hh}} - G_{B_{hr}} + G_{B_{rt}} - G_{B_{rh}} + G_{B_{rr}}$ 。

16. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法, 其特征在于, 在步骤 (5) 中所述的乘积二的计算方法, 当边角类型为 a 时, 为 $G_{D_{Ar_a}} = G_{D_{Zero}} - G_{D_{th}} + G_{D_{tr}} + G_{D_{rt}} - G_{D_{ht}} - G_{D_{hh}} + G_{D_{hl}} + G_{D_{lh}} - G_{D_{ll}} + G_{D_{pp}}$ 。

17. 根据权利要求 1 所述的 AR 边界条件下的图像恢复替代计算方法, 其特征在于, 在步骤 (5) 中所述的乘积二的计算方法, 当边角类型为 b 时, 为 $G_{D_{Ar_b}} = G_{D_{Zero}} - G_{D_{th}} + G_{D_{tr}} - G_{D_{ht}} + G_{D_{hh}} - G_{D_{hr}} + G_{D_{rt}} - G_{D_{rh}} + G_{D_{rr}}$ 。

AR 边界条件下图像模糊矩阵与矢量乘积的替代计算方法

技术领域

[0001] 本发明涉及图像处理。特别地,本发明涉及 AR 边界下,大型图像模糊矩阵与图像矢量乘积的替代计算。

背景技术

[0002] 在图像滤波中需要计算模糊矩阵与图像矢量的乘积 Kf (称为乘积一), 在图像恢复中需要计算模糊矩阵及其转置与图像矢量的乘积 $K'Kf$ (将 $K'f$ 称为乘积二), (其中 $K \in R^{m \times m}$ 为点扩展函数 $K \in R^{p \times q}$ 的模糊矩阵, $f \in R^{m \times 1}$ 为图像 $F \in R^{m \times n}$ 的矢量表示)。 K 和 K' 为大型的稀疏矩阵, 使乘积一和乘积二不能直接计算, 对此, 目前主要两种处理方式: 一种是基于预置矩阵的加速方式, 另一种是基于卷积或模糊矩阵对角化的替代计算方法。当模糊矩阵过大时, 预置矩阵也过大, 使加速方式无法再采用, 而此时替代计算能否可用, 以及具体采用何种替代方式, 与图像滤波和图像恢复所基于的图像边界条件 (boundary conditions, BCs) 类型有关。

[0003] 图像边界类型中, 传统的有 Zero BCs、Periodic BCs、Neumann BCs, 新近刚被提出的有 Anti-Reflective (AR) BCs 和外推 (又称平均) BCs。其中, 基于 AR BCs 的图像滤波与图像恢复, 可获得较为自然的图像边界, 但对应的模糊矩阵结构复杂, 无法直接用卷积或模糊矩阵对角化的方法替代, 对此, 目前还没有出现有效的解决办法。由此, 需要寻找一种能有效处理 AR BCs 下乘积一和乘积二的替代计算方法。

发明内容

[0004] 本发明所要解决的技术问题是: 在 AR BCs 下, 无论边角类型为 a 还是 b, 对应大型模糊矩阵都不带可利用结构, 使的乘积一和乘积二既不能直接计算, 也不能直接用卷积方法或对角化方法替代。

[0005] 针对所述问题, 本发明在图 1 中, 给出了一种替代计算方法, 具体为: 在 AR BCs 下, 根据边角类型, 先将乘积一和乘积二, 分解为多个矩阵与矢量乘积的和, 保证所产生的各个分解部分的矩阵, 都带可利用的分块结构; 利用各个分块矩阵的结构, 构造其卷积核; 各个分块矩阵与图像矢量的乘积, 用对应卷积核与原始图像某个边界部分间的卷积来实现; 综合各个卷积结果, 形成乘积一和乘积二的结果图像, 从而替代地计算了乘积一和乘积二。

[0006] 为支持上述替代计算方法, 本发明的各个实施例中给出了具体计算方法, 包括:

[0007] 当边角为 a 类型时 (四个边角外的像素, 直接由边角内反对称的像素计算), 本发明的一些实施例, 将乘积一和乘积二分别分解为多个矩阵与图像矢量的乘积, 使得分解后的各个矩阵, 带可利用的分块结构。

[0008] 当边角为 b 类型 (四个边角外的像素, 先行方向反对称, 再列方向反对称) 时, 本发明一些实施例, 将乘积一和乘积二作类似的分解, 使得分解后各个矩阵带可利用的分块结构。

[0009] 当边角类型为 a 时, 一些实施例构造乘积一中各分解模糊矩阵的点扩展函数。

- [0010] 当边角类型为 b 时,一些实施例构造乘积一中各分解模糊矩阵的点扩展函数。
- [0011] 当边角类型为 a 时,一些实施例构造乘积二中各分解模糊矩阵的点扩展函数。
- [0012] 当边角类型为 b 时,一些实施例构造乘积二中各分解模糊矩阵的点扩展函数。
- [0013] 当边角类型为 a 时,一些实施例基于卷积,计算乘积一中的边界部分。
- [0014] 当边角类型为 b 时,一些实施例基于卷积,计算乘积一中的边界部分。
- [0015] 当边角类型为 a 时,一些实施例基于卷积,计算乘积二中的边界部分。
- [0016] 当边角类型为 b 时,一些实施例基于卷积,计算乘积二中的边界部分。
- [0017] 一些实例计算 Zero BCs 下的乘积一。
- [0018] 一些实例计算 Zero BCs 下的乘积二。
- [0019] 一些实施例在 AR BCs 下,当边角类型为 a 时,计算乘积一。
- [0020] 一些实施例在 AR BCs 下,当边角类型为 b 时,计算乘积一。
- [0021] 一些实施例在 AR BCs 下,当边角类型为 a 时,计算乘积二。
- [0022] 一些实施例在 AR BCs 下,当边角类型为 b 时,计算乘积二。
- [0023] 一些实施例通过实验,验证了本发明所给出替代计算方法的有效性。
- [0024] 本发明所给出的替代计算方法,解决了 AR BCs 下当边角为任意一种类型时,乘积一和乘积二的计算难题,可集成到 AR BCs 下的图像滤波和图像恢复中,具有应用价值。

附图说明

- [0025] 在随附的权利要求中阐述了本发明的新颖特征。出于说明的目的,以下附图辅助阐述本发明的相关实施例。
- [0026] 图 1 示出的是本发明的实施流程。
- [0027] 图 2 示出的是原始图像 Cameraman。
- [0028] 图 3 示出的是 Zero BCs 下的乘积一的计算结果。
- [0029] 图 4 示出的是采用原始图像作为参考图像,在 AR BCs 下,当边角为 a 类型时,乘积一的替代计算结果。
- [0030] 图 5 示出的是采用原始图像作为参考图像,在 AR BCs 下,当边角为 a 类型时,乘积二的替代计算结果。
- [0031] 图 6 示出的是采用原始图像作为参考图像,在 AR BCs 下,当边角为 b 类型时,乘积一的替代计算。
- [0032] 图 7 示出的是采用原始图像作为参考图像,在 AR BCs 下,当边角为 b 类型时,乘积二的替代计算结果。
- [0033] 图 8 示出的是采用 AR BCs 下, a 边角类型时的退化图像作为参考图像,在相同边界和边角下,乘积一的替代计算结果。
- [0034] 图 9 示出的是采用 AR BCs 下, a 边角类型时的退化图像作为参考图像,在相同边界和边角下,乘积二的替代计算结果。
- [0035] 图 10 示出的是采用 AR BCs 下, b 边角类型时的退化图像作为参考图像,在相同边界和边角下,乘积一的替代计算结果。
- [0036] 图 11 示出的是采用 AR BCs 下, b 边角类型时的退化图像作为参考图像,在相同边界和边角下,乘积二的替代计算结果。

[0037] 表 2 示出的是分别基于真实图像和退化图像,采用本发明的替代计算得到的乘积一和乘积二,与模糊矩阵与真实图像矢量乘积得到的结果(作为参照)间的平方根误差(rmse),以及直接将 Zero BCs 下的乘积一和乘积二,当作 AR BCs 下的乘积一和乘积二,所产生的 rmse。rmse 的计算方法为 $rmse = \text{norm}(\text{img_Real} - \text{img_P}) / \sqrt{\text{length}(\text{img_M}(:))}$,其中 img_Real 由模糊矩阵与真实图像矢量间的乘积得到(计算 $H * f$, 或 $H' * f$),img_P 为采用本专利的替代计算结果,以及 Zero BCs 下的计算结果。norm() 为 2-范数。

[0038] 表 3 示出的是在不同边角条件下,用直接用模糊矩阵与图像矢量乘积计算乘积一和乘积二,与本发明替代计算乘积一和乘积二,分别耗费的时间。

具体实施方式

[0039] 出于说明的目的,在以下描述中,对发明内容中各个实施例中的具体计算方法进行阐述。为了表达清晰和直接编程实现,计算式公式中应用了一些 Matlab 的函数形式表示。

[0040] 第 1 实施例

[0041] 当边角类型为 a 时,本发明给出乘积一的分解方法,具体为:将乘积一 $g = Kf$,分解为 $g_{B_Ar_a} = (K_{B_tt} - K_{B_th} + K_{B_tr} - K_{B_ht} - K_{B_hh} + K_{B_rt} + K_{B_lh} + K_{B_hl} - K_{B_ll} + K_{B_pp})f$,使得中心部分 $K_{B_tt}f$ 对应的是 Zero BCs 下的乘积一,其他则对应的是乘积一的边界部分。

[0042] 分解矩阵的各下标中,B 表示乘积一,第 2 个下标表示块间结构,第 3 个下标表示块内结构,t 表示 Toeplitz 矩阵,h 表示 Hankel 矩阵,l 表示取对应 Hankel 矩阵的第 1 列和最后一列,p 表示秩 -2 的修正阵,r 表示另一种秩 -2 的修正阵(这里不详细展开 p 结构和 r 结构的具体形式,这不影响后续方法的描述)。各分解矩阵带可利用的分块结构,便于设计对应的点扩展函数。

[0043] 第 2 实施例

[0044] 当边角类型为 b 时,本发明将乘积一,分解为 $g_{B_Ar_b} = (K_{B_tt} - K_{B_th} + K_{B_tr} - K_{B_ht} + K_{B_hh} - K_{B_hr} + K_{B_rt} - K_{B_rh} + K_{B_rr})f$,其中各个分块矩阵的小写下标含义如第 1 实施例。

[0045] 第 3 实施例

[0046] 当边角类型为 a 时,本发明将乘积二 $g = K'f$,分解为 $g_{D_Ar_a} = (K_{D_tt} - K_{D_th} + K_{D_tr} - K_{D_ht} - K_{D_hh} + K_{D_rt} + K_{D_lh} + K_{D_hl} - K_{D_ll} + K_{D_pp})f$,各矩阵的第 1 个下标 D 表示乘积二。

[0047] 第 4 实施例

[0048] 当边角类型为 b 时,本发明将乘积二分解为: $g_{D_Ar_b} = (K_{D_tt} - K_{D_th} + K_{D_tr} - K_{D_ht} + K_{D_hh} - K_{D_hr} + K_{D_rt} - K_{D_rh} + K_{D_rr})f$ 。

[0049] 第 5 实施例

[0050] 合称模糊矩阵 K 及其转置 K' 为变换矩阵,基于第 1 到第 4 实施例,给出 AR BCs 下关于乘积一和乘积二的统一计算流程(流程图如图 1 所示),具体包括:

[0051] (1) 输入边角类型;

[0052] (2) 根据边角类型,计算各个分解变换矩阵的点扩展函数;

[0053] (3) 根据边角类型,在步骤(2)的基础上,基于卷积,计算变换图像边界部分的矩阵形式;

[0054] (4) 计算 Zero BCs 下的变换图像;

[0055] (5) 根据边角类型, 计算 AR BCs 下的变换图像。

[0056] 后续的各个实施例, 是对第 5 实施例中第 2 到第 5 步的具体说明。

[0057] 第 6 实施例

[0058] 当边角类型为 a 时, 本发明给出第 1 实施例中 K_{B_th} 、 K_{B_tr} 、 K_{B_ht} 、 K_{B_hh} 、 K_{B_rt} 、 K_{B_lh} 、 K_{B_hl} 、 K_{B_ll} 和 K_{B_pp} 的点扩展函数。具体为:

[0059] 记 $p_1 = \lfloor p/2 \rfloor$, $q_1 = \lfloor q/2 \rfloor$, $\text{fliplr}(\cdot)$ 表示对矩阵列方向 (左右) 翻转, $\text{flipud}(\cdot)$ 表示对矩阵行方向 (上下) 翻转, $\text{conv2}(\cdot, \cdot)$ 表示二维卷积, $i:j$ 表示从 i 到 j 的所有下标, $V(:, :)$ 中的 $:$ 标引矩阵 V 的所有行与列, $V(\text{end}, \text{end})$ 中的 end 标引对应矩阵的最后行与最后一列。则:

[0060] (1) 对 K_{B_th} , 构造 2 个点扩展函数 $k_{B_th_1} = \text{fliplr}(k(:, q_1+1:q))$, $k_{B_th_2} = \text{fliplr}(k(:, 1:q_1+1))$;

[0061] (2) K_{B_tr} 的 2 个大小为 $p \times (q_1+1)$ 的点扩展函数的元素, 分别由

$$k_{B_tr_1}(i, j) = k(i, q_1+j) + 2 \sum_{s_1=q_1+1+j}^q k(i, s_1), 1 \leq j \leq q_1, \quad k_{B_tr_1}(i, q_1+1) = k(i, q_1+1) \text{ 和}$$

$$k_{B_tr_2}(i, j) = k(i, j) + 2 \sum_{s_1=1}^{j-1} k(i, s_1), 2 \leq j \leq q_1+1, \quad k_{B_tr_2}(i, 1) = k(i, 1) \text{ 计算, 其中 } i = 1, \dots, p;$$

[0062] (3) 对 K_{B_ht} , 构造 2 个点扩展函数 $k_{B_ht_1} = \text{flipud}(k(p_1+1:p, :))$, $k_{B_ht_2} = \text{flipud}(k(1:p_1+1, :))$;

[0063] (4) 对 K_{B_hh} , 构造 4 个点扩展函数 $k_{B_hh_1} = \text{fliplr}(\text{flipud}(k(p_1+1:p, q_1+1:q)))$, $k_{B_hh_2} = \text{fliplr}(\text{flipud}(k(p_1+1:p, 1:q_1+1)))$, $k_{B_hh_3} = \text{fliplr}(\text{flipud}(k(1:p_1+1, q_1+1:q)))$, $k_{B_hh_4} = \text{fliplr}(\text{flipud}(k(1:p_1+1, 1:q_1+1)))$;

[0064] (5) 对 K_{B_rt} , 构造 2 个点扩展函数

$$k_{B_rt_1}(i, :) = k(p_1+2-i, :) + 2 \sum_{s_1=p_1+1+i}^p k(s_1, :), 1 \leq i \leq p_1, \quad k_{B_rt_1}(p_1+1, :) = k(p, :) \text{ 和}$$

$$k_{B_rt_2}(i, :) = k(i, :) + 2 \sum_{s_1=1}^{i-1} k(s_1, :), 2 \leq i \leq p_1+1, \quad k_{B_rt_2}(1, :) = k(1, :);$$

[0065] (6) K_{B_lh} 与 K_{B_hh} 的点扩展函数相同, 即 $k_{B_lh_i} = k_{B_hh_i}$, $i = 1, \dots, 4$;

[0066] (7) K_{B_hl} 与 K_{B_hh} 的点扩展函数相同, 即 $k_{B_hl_i} = k_{B_hh_i}$, $i = 1, \dots, 4$;

[0067] (8) K_{B_ll} 与 K_{B_hh} 的点扩展函数相同, 即 $k_{B_ll_i} = k_{B_hh_i}$, $i = 1, \dots, 4$;

[0068] (9) K_{B_pp} 的 4 个点扩展函数 $k_{B_pp_t}$, $t = 1, \dots, 4$, 分别由公式

$$k_{B_pp_1}(i, j) = 2 \sum_{s_1=1}^{p_1+1-i} \sum_{s_2=1}^{q_1+1-j} k(s_1, s_2), \quad k_{B_pp_2}(i, j) = 2 \sum_{s_1=1}^{p_1+1-i} \sum_{s_2=q_1+1-j}^q k(s_1, s_2),$$

$$k_{B_pp_3}(i, j) = 2 \sum_{s_1=p_1+1+i}^p \sum_{s_2=1}^{q_1+1-j} k(s_1, s_2), \text{ 和}$$

$$k_{B_pp_4}(i, j) = 2 \sum_{s_1=p_1+1+i}^p \sum_{s_2=q_1+1+j}^q k(s_1, s_2), 1 \leq i \leq p_1, 1 \leq j \leq q_1 \text{ 构造。}$$

[0069] 第 7 实施例

[0070] 当边角为 b 类型时, 本发明给出第 2 实施例中, 除 K_{B_th} 、 K_{B_tr} 、 K_{B_ht} 、 K_{B_hh} 、 K_{B_rt} 外其他三个分解矩阵 K_{B_hr} 、 K_{B_rh} 和 K_{B_rr} 等的点扩展函数。具体为:

[0071] (1) $K_{B_{hr}}$ 的 4 个点扩展函数与 $K_{B_{tr}}$ 点扩展函数相关, 具体为: $k_{B_{hr_1}} = \text{flipud}(k_{B_{tr_1}}(1:p_1+1, :))$, $k_{B_{hr_2}} = \text{flipud}(k_{B_{tr_2}}(p_1+1:\text{end}, :))$, $k_{B_{hr_3}} = \text{flipud}(k_{B_{tr_1}}(\text{end}-p_1:\text{end}, :))$, $k_{B_{hr_4}} = \text{flipud}(k_{B_{tr_2}}(1:p_1+1, :))$;

[0072] (2) $K_{B_{rh}}$ 的 4 个点扩展函数与 $K_{B_{rt}}$ 的点扩展函数相关, 具体为:

[0073] $k_{B_{rh_1}} = \text{fliplr}(\text{flipud}(k_{B_{rt_1}}(1:p_1+1, q_1+1:\text{end})))$,

[0074] $k_{B_{rh_2}} = \text{fliplr}(\text{flipud}(k_{B_{rt_1}}(1:p_1+1, 1:q_1+1)))$, $k_{B_{rh_3}} = \text{flipud}(k_{B_{rt_2}}(1:p_1+1, \text{end}-q_1:\text{end}))$,

[0075] $k_{B_{rh_4}} = \text{flipud}(k_{B_{rt_2}}(1:p_1+1, 1:q_1+1))$;

[0076] (3) $K_{B_{rr}}$ 的 4 点扩展函数与 $k_{B_{tr_1}}$ 和 $k_{B_{tr_2}}$ 相关, 计算方法分别为:

[0077] $k_{B_{rr_1}}(i,:) = k_{B_{tr_1}}(p_1+i,:) + 2 \sum_{s_1=p_1+i+1}^p k_{B_{tr_1}}(s_1, :)$, $1 \leq i \leq p_1$, $k_{B_{rr_1}}(p_1+1, :) = k_{B_{tr_1}}(p, :)$;

[0078] $k_{B_{rr_2}}(i,:) = k_{B_{tr_2}}(p_1+i,:) + 2 \sum_{s_1=p_1+i+1}^p k_{B_{tr_2}}(s_1, :)$, $1 \leq i \leq p_1$, $k_{B_{rr_2}}(p_1+1, :) = k_{B_{tr_2}}(p, :)$;

[0079] $k_{B_{rr_3}}(i,:) = k_{B_{tr_1}}(i,:) + 2 \sum_{s_1=1}^{i-1} k_{B_{tr_1}}(s_1, :)$, $2 \leq i \leq p_1+1$, $k_{B_{rr_3}}(1, :) = k_{B_{tr_1}}(1, :)$;

[0080] $k_{B_{rr_4}}(i,:) = k_{B_{tr_2}}(i,:) + 2 \sum_{s_1=1}^{i-1} k_{B_{tr_2}}(s_1, :)$, $2 \leq i \leq p_1+1$, $k_{B_{rr_4}}(1, :) = k_{B_{tr_2}}(1, :)$ 。

[0081] 第 8 实施例

[0082] 当边角为 a 类型时, 本发明给出第 3 实施例中矩阵 $K_{D_{th}}$ 、 $K_{D_{tr}}$ 、 $K_{D_{ht}}$ 、 $K_{D_{hh}}$ 、 $K_{D_{rt}}$ 、 $K_{D_{lh}}$ 、 $K_{D_{hl}}$ 、 $K_{D_{ll}}$ 和 $K_{D_{pp}}$ 等的点扩展函数。具体为:

[0083] (1) $K_{D_{th}}$ 的点扩展函数与实例中 $K_{B_{th}}$ 的点扩展函数相关, 具体为:

[0084] $k_{D_{th_i}} = \text{flipud}(k_{B_{th_i}})$, $i = 1, 2$;

[0085] (2) $K_{D_{tr}}$ 的点扩展函数与实例中 $K_{B_{tr}}$ 的点扩展函数相关, 具体为:

[0086] $k_{D_{tr_i}} = \text{flipud}(k_{B_{tr_i}})$, $i = 1, 2$;

[0087] (3) $K_{D_{ht}}$ 的点扩展函数与实例中 $K_{B_{ht}}$ 的点扩展函数相关, 具体为:

[0088] $k_{D_{ht_i}} = \text{fliplr}(k_{B_{ht_i}})$, $i = 1, 2$;

[0089] (4) $K_{D_{hh}}$ 的点扩展函数与 $K_{B_{hh}}$ 的相同, 即 $k_{D_{hh_i}} = k_{B_{hh_i}}$, $i = 1, \dots, 4$;

[0090] (5) 对 $K_{D_{rt}}$ 的点扩展函数与 $K_{B_{rt}}$ 的相关, 具体为: 的构造 2 个点扩展函数

[0091] $k_{D_{rt_i}} = \text{fliplr}(\text{flipud}(k_{B_{rt_i}}))$, $i = 1, 2$;

[0092] (6) $K_{D_{lh}}$ 与 $K_{B_{hh}}$ 的点扩展函数相同, 即 $k_{D_{lh_i}} = k_{B_{hh_i}}$, $i = 1, \dots, 4$;

[0093] (7) $K_{D_{hl}}$ 与 $K_{B_{hl}}$ 的点扩展函数相同, 即 $k_{D_{hl_i}} = k_{B_{hl_i}}$, $i = 1, \dots, 4$;

[0094] (8) $K_{D_{ll}}$ 与 $K_{B_{ll}}$ 的点扩展函数相同, 即 $k_{D_{ll_i}} = k_{B_{ll_i}}$, $i = 1, \dots, 4$;

[0095] (9) $K_{D_{pp}}$ 的点扩展函数与 $K_{B_{pp}}$ 的相同, 即 $k_{D_{pp_i}} = k_{B_{pp_i}}$, $i = 1, \dots, 4$ 。

[0096] 第 9 实施例

[0097] 当边角为 b 类型时, 本发明给出第四实施例中, 除 $K_{D_{th}}$ 、 $K_{D_{tr}}$ 、 $K_{D_{ht}}$ 、 $K_{D_{hh}}$ 、 $K_{D_{rt}}$ 外的其他三个分解矩阵 $K_{D_{hr}}$ 、 $K_{D_{rh}}$ 和 $K_{D_{rr}}$ 等的点扩展函数。具体为:

[0098] (1) $K_{D_{hr}}$ 的 4 个点扩展函数与 $K_{B_{hr}}$ 点扩展函数相关, 具体为: $k_{D_{hr_i}} = \text{fliplr}(k_{B_{hr_i}})$, $i = 1, \dots, 4$;

[0099] (2) $K_{D_{rh}}$ 的 4 个点扩展函数与 $K_{B_{rh}}$ 的点扩展函数相同, 具体为: $k_{D_{rh_i}} = k_{B_{rh_i}}$, $i =$

1, ..., 4;

[0100] (3) $K_{D_{rr}}$ 的点扩展函数与 $K_{B_{rr}}$ 的相关, 即: $k_{D_{rr}, i} = \text{fliplr}(\text{flipud}(k_{B_{rr}, i}))$, $i = 1, \dots, 4$ 。

[0101] 第 10 实施例

[0102] 记第 1 实施例的公式 $g_{B_{Ar}, a} = (K_{B_{tt}} - K_{B_{th}} + K_{B_{B_{tr}}} - K_{B_{ht}} - K_{B_{hh}} + K_{B_{rt}} + K_{B_{lh}} + K_{B_{hl}} - K_{B_{ll}} + K_{B_{pp}})f$ 中, 除 $K_{B_{tt}}$ 外其他矩阵矢量乘积的矩阵结果 (大小为 $m \times n$), 分别为 $G_{B_{th}}, G_{B_{tr}}, G_{B_{rt}}, G_{B_{ht}}, G_{B_{hh}}, G_{B_{hl}}, G_{B_{lh}}, G_{B_{ll}}$ 和 $G_{B_{pp}}$, 本发明给出这些矩阵计算方法, 具体为:

[0103] (1) 对 $G_{B_{th}}$, 计算 $r_{B_{th}, 1} = \text{conv}(k_{B_{th}, 1}, F(:, 1:q_1+1))$, $r_{B_{th}, 2} = \text{conv}(k_{B_{th}, 2}, F(:, n-q_1:n))$, 则 $G_{B_{th}}(:, 1:q_1+1) = \text{fliplr}(r_{B_{th}, 1}(p_1+1:\text{end}-p_1, 1:q_1+1))$, $G_{B_{th}}(:, n-q_1:n) = \text{fliplr}(r_{B_{th}, 2}(p_1+1:\text{end}-p_1, q_1+1:\text{end}))$, $G_{B_{th}}$ 的其他元素为 0;

[0104] (2) 对 $G_{B_{tr}}$, 计算 $r_{B_{tr}, 1} = \text{conv2}(\text{fliplr}(k_{B_{tr}, 1}), F(:, 1))$, $r_{B_{tr}, 2} = \text{conv2}(\text{fliplr}(k_{B_{tr}, 2}), F(:, n))$, $G_{B_{tr}}(:, 1:q_1+1) = \text{fliplr}(r_{B_{tr}, 1}(p_1+1:\text{end}-p_1, :))$, $G_{B_{tr}}(:, n-q_1:n) = \text{fliplr}(r_{B_{tr}, 2}(p_1+1:\text{end}-p_1, :))$, $G_{B_{tr}}$ 的其他元素为 0;

[0105] (3) 对 $G_{B_{ht}}$, 计算 $r_{B_{ht}, 1} = \text{conv2}(k_{B_{ht}, 1}, F(1:p_1+1, :))$, $r_{B_{ht}, 2} = \text{conv2}(k_{B_{ht}, 2}, F(m-p_1:m, :))$, 则 $G_{B_{ht}}(1:p_1+1, :) = \text{flipud}(r_{B_{ht}, 1}(1:p_1+1, q_1+1:\text{end}-q_1))$, $G_{B_{ht}}(m-p_1:m, :) = \text{flipud}(r_{B_{ht}, 2}(p_1+1:\text{end}, q_1+1:\text{end}-q_1))$, $G_{B_{ht}}$ 的其他元素为 0;

[0106] (4) 对 $G_{B_{hh}}$, 计算 $r_{B_{hh}, 1} = \text{conv2}(k_{B_{hh}, 1}, F(1:p_1+1, 1:q_1+1))$, $r_{B_{hh}, 2} = \text{conv2}(k_{B_{hh}, 2}, F(1:p_1+1, n-q_1:n))$, $r_{B_{hh}, 3} = \text{conv2}(k_{B_{hh}, 3}, F(m-p_1:m, 1:q_1+1))$, $r_{B_{hh}, 4} = \text{conv2}(k_{B_{hh}, 4}, F(m-p_1:m, n-q_1:n))$, 则 $G_{B_{hh}}(1:p_1+1, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{hh}, 1}(1:p_1+1, 1:q_1+1)))$, $G_{B_{hh}}(1:p_1+1, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{hh}, 2}(1:p_1+1, \text{end}-q_1:\text{end})))$, $G_{B_{hh}}(m-p_1:m, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{hh}, 3}(\text{end}-p_1:\text{end}, 1:q_1+1)))$, $G_{B_{hh}}(m-p_1:m, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{hh}, 4}(\text{end}-p_1:\text{end}, 1:q_1+1)))$, $G_{B_{hh}}(m-p_1:m, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{hh}, 4}(\text{end}-p_1:\text{end}, \text{end}-q_1:\text{end})))$, $G_{B_{hh}}$ 的其他元素为 0;

[0107] (5) 对 $G_{B_{rt}}$, 计算 $r_{B_{rt}, 1} = \text{conv2}(k_{B_{rt}, 1}, F(1, :))$, $r_{B_{rt}, 2} = \text{conv2}(k_{B_{rt}, 2}, F(1, :))$, $r_{B_{rt}, 1} = \text{conv2}(k_{B_{rt}, 1}(1:p_1+1), F(1, :))$, $r_{B_{rt}, 2} = \text{conv2}(\text{flipud}(k_{B_{rt}, 2}(1:p_1+1)), F(m, :))$, 则 $G_{B_{rt}}(1:p_1+1, :) = r_{B_{rt}, 1}(:, q_1+1:\text{end}-q_1)$, $G_{B_{rt}}(m-p_1:m, :) = r_{B_{rt}, 2}(:, q_1+1:\text{end}-q_1)$, $G_{B_{rt}}$ 的其他元素为 0;

[0108] (6) 对 $G_{B_{lh}}$, 计算 $r_{B_{lh}, 1} = \text{conv2}(k_{B_{lh}, 1}, F(1, 1:q_1+1))$, $r_{B_{lh}, 2} = \text{conv2}(k_{B_{lh}, 2}, F(1, n-q_1:n))$, $r_{B_{lh}, 3} = \text{conv2}(k_{B_{lh}, 3}, F(m, 1:q_1+1))$, $r_{B_{lh}, 4} = \text{conv2}(k_{B_{lh}, 4}, F(m, n-q_1:n))$, 则 $G_{B_{lh}}(1:p_1+1, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{lh}, 1}(1:p_1+1, 1:q_1+1)))$, $G_{B_{lh}}(1:p_1+1, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{lh}, 2}(1:p_1+1, \text{end}-q_1:\text{end})))$, $G_{B_{lh}}(m-p_1:m, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{lh}, 3}(\text{end}-p_1:\text{end}, 1:q_1+1)))$, $G_{B_{lh}}(m-p_1:m, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{lh}, 4}(1:\text{end}, q_1+1:\text{end})))$, $G_{B_{lh}}$ 的其他元素为 0;

[0109] (7) 对于 $G_{B_{hl}}$, 计算 $r_{B_{hl}, 1} = \text{conv2}(k_{B_{hl}, 1}, F(1:p_1+1, 1))$, $r_{B_{hl}, 2} = \text{conv2}(k_{B_{hl}, 2}, F(1:p_1+1, n))$, $r_{B_{hl}, 3} = \text{conv2}(k_{B_{hl}, 3}, F(m-p_1:m, 1))$, $r_{B_{hl}, 4} = \text{conv2}(k_{B_{hl}, 4}, F(m-p_1:m, n))$, 则 $G_{B_{hl}}(1:p_1+1, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{hl}, 1}(1:p_1+1, 1:q_1+1)))$, $G_{B_{hl}}(1:p_1+1, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{hl}, 2}(1:p_1+1, \text{end}-q_1:\text{end})))$, $G_{B_{hl}}(m-p_1:m, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{hl}, 3}(\text{end}-p_1:\text{end}, \text{end}-q_1:\text{end})))$, $G_{B_{hl}}(m-p_1:m, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{hl}, 4}(\text{end}-p_1:\text{end}, :)))$, $G_{B_{hl}}$ 的其他元素为 0;

[0110] (8) 对 $G_{B_{11}}$, 计算 $r_{B_{11_1}} = \text{conv2}(k_{11_1}, F(1, 1))$, $r_{B_{11_2}} = \text{conv2}(k_{B_{11_2}}, F(1, n))$, $r_{B_{11_3}} = \text{conv2}(k_{B_{11_3}}, F(m, 1))$, $r_{B_{11_4}} = \text{conv2}(k_{B_{11_4}}, F(m, n))$, 则 $G_{B_{11}}(1:p_1+1, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{11_1}}(1:p_1+1, 1:q_1+1)))$, $G_{B_{11}}(1:p_1+1, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{11_2}}(1:p_1+1, \text{end}-q_1:\text{end})))$, $G_{B_{11}}(m-p_1:m, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{11_3}}))$, $G_{B_{11}}(m-p_1:m, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{11_4}}))$, $G_{B_{11}}$ 的其他元素为 0;

[0111] (9) 对 $G_{B_{pp}}$, 计算 $r_{B_{pp_1}} = \text{conv2}(\text{fliplr}(\text{flipud}(k_{B_{pp_1}})), F(m, n))$, $r_{B_{pp_2}} = \text{conv2}(\text{fliplr}(\text{flipud}(k_{B_{pp_2}})), F(m, 1))$, $r_{B_{pp_3}} = \text{conv2}(\text{fliplr}(k_{B_{pp_3}}), F(1, n))$, $r_{B_{pp_4}} = \text{conv2}(k_{B_{pp_4}}, F(1, 1))$, 则 $G_{B_{pp}}(m-p_1+1:m, n-q_1+1:n) = r_{B_{pp_1}}$, $G_{B_{pp}}(m-p_1+1:m, 1:q_1) = \text{fliplr}(r_{B_{pp_2}})$, $G_{B_{pp}}(1:p_1, n-q_1+1:n) = r_{B_{pp_3}}$, $G_{B_{pp}}(1:p_1, 1:q_1) = r_{B_{pp_4}}$, $G_{B_{pp}}$ 的其他元素为 0。

[0112] 第 11 实施例

[0113] 记第 2 实施例的公式 $g_{B_{Ar-b}} = (K_{B_{tt}} - K_{B_{th}} + K_{B_{tr}} - K_{B_{ht}} + K_{B_{hh}} - K_{B_{hr}} + K_{B_{rt}} - K_{B_{rh}} + K_{B_{rr}})f$ 中, 除 $K_{B_{tt}}f$ 外其他各个模糊矩阵与矢量乘积的矩阵结果 (大小为 $m \times n$) 分别为 $G_{B_{th}}$ 、 $G_{B_{tr}}$ 、 $G_{B_{rt}}$ 、 $G_{B_{ht}}$ 和 $G_{B_{hh}}$ 、 $G_{B_{hr}}$ 、 $G_{B_{rh}}$ 和 $G_{B_{rr}}$, 本发明给出都应矩阵的计算方法。具体为:

[0114] (1) $G_{B_{th}}$ 、 $G_{B_{tr}}$ 、 $G_{B_{rt}}$ 、 $G_{B_{ht}}$ 和 $G_{B_{hh}}$ 等的计算方法分别与第 10 实施例中同名矩阵的计算方法相同;

[0115] (2) 对 $G_{B_{hr}}$, 计算 $r_{B_{hr_1}} = \text{conv2}(k_{B_{hr_1}}, F(1:p_1+1, 1))$, $r_{B_{hr_2}} = \text{conv2}(k_{B_{hr_2}}, F(1:p_1+1, n))$, $r_{B_{hr_3}} = \text{conv2}(k_{B_{hr_3}}, F(m-p_1:m, 1))$, $r_{B_{hr_4}} = \text{conv2}(k_{B_{hr_4}}, F(m-p_1:m, n))$, $G_{B_{hr}}(1:p_1+1, 1:q_1+1) = \text{flipud}(r_{B_{hr_1}}(1:p_1+1, :))$, $G_{B_{hr}}(1:p_1+1, n-q_1:n) = \text{flipud}(r_{B_{hr_2}}(1:p_1+1, :))$, $G_{B_{hr}}(m-p_1:m, 1:q_1+1) = \text{flipud}(r_{B_{hr_3}}(\text{end}-p_1:\text{end}, :))$, $G_{B_{hr}}(m-p_1:m, n-q_1:n) = \text{flipud}(r_{B_{hr_4}}(\text{end}-p_1:\text{end}, :))$, $G_{B_{hr}}$ 的其他元素为 0;

[0116] (3) 对 $G_{B_{rh}}$, 计算 $r_{B_{rh_1}} = \text{conv2}(k_{B_{rh_1}}, F(1, 1:q_1+1))$, $r_{B_{rh_2}} = \text{conv2}(k_{B_{rh_2}}, F(1, n-q_1:n))$, $r_{B_{rh_3}} = \text{conv2}(k_{B_{rh_3}}, F(m, 1:q_1+1))$, $r_{B_{rh_4}} = \text{conv2}(k_{B_{rh_4}}, F(m, n-q_1:n))$, 则 $G_{B_{rh}}(1:p_1+1, 1:q_1+1) = \text{fliplr}(\text{flipud}(r_{B_{rh_1}}(:, 1:q_1+1)))$, $G_{B_{rh}}(1:p_1+1, n-q_1:n) = \text{fliplr}(\text{flipud}(r_{B_{rh_2}}(:, \text{end}-q_1:\text{end})))$, $G_{B_{rh}}(m-q_1:m, 1:q_1+1) = \text{fliplr}(r_{B_{rh_3}}(:, 1:q_1+1))$, $G_{B_{rh}}(m-p_1:m, n-q_1:n) = \text{fliplr}(r_{B_{rh_4}}(:, \text{end}-q_1:\text{end}))$, $G_{B_{rh}}$ 的其他元素为 0;

[0117] (4) 对 $G_{B_{rr}}$, 计算 $r_{B_{rr_1}} = \text{conv2}(\text{fliplr}(k_{B_{rr_1}}), F(1, 1))$, $r_{B_{rr_2}} = \text{conv2}(\text{fliplr}(k_{B_{rr_2}}), F(1, n))$, $r_{B_{rr_3}} = \text{conv2}(k_{B_{rr_3}}, F(m, 1))$, $r_{B_{rr_4}} = \text{conv2}(k_{B_{rr_4}}, F(m, n))$ 则 $G_{B_{rr}}(1:p_1+1, 1:q_1+1) = \text{fliplr}(r_{B_{rr_1}})$, $G_{B_{rr}}(1:p_1+1, n-q_1+1:n) = \text{fliplr}(r_{B_{rr_2}})$, $G_{B_{rr}}(m-p_1+1:m, 1:q_1+1) = r_{B_{rr_3}}$, $G_{B_{rr}}(m-p_1+1:m, n-q_1+1:n) = r_{B_{rr_4}}$, $G_{B_{rr}}$ 的其他元素为 0。

[0118] 第 12 实施例

[0119] 记第 3 实施例的公式 $g_{D_{Ar-a}} = (K_{D_{tt}} - K_{D_{th}} + K_{D_{tr}} - K_{D_{ht}} - K_{D_{hh}} + K_{D_{rt}} + K_{D_{lh}} + K_{D_{hl}} - K_{D_{ll}} + K_{D_{pp}})f$ 中, 除 $K_{D_{tt}}f$ 外其他各个模糊矩阵与图像矢量乘积的矩阵结果 (大小为 $m \times n$) 分别为 $G_{D_{th}}$ 、 $G_{D_{tr}}$ 、 $G_{D_{rt}}$ 、 $G_{D_{ht}}$ 、 $G_{D_{hh}}$ 、 $G_{D_{hl}}$ 、 $G_{D_{lh}}$ 、 $G_{D_{ll}}$ 和 $G_{D_{pp}}$, 本发明给出各矩阵的计算方法, 具体为:

[0120] (1) 对 $G_{D_{th}}$, 计算 $r_{D_{th_1}} = \text{conv}(k_{D_{th_1}}, F(:, 1:q_1+1))$, $r_{D_{th_2}} = \text{conv}(k_{D_{th_2}}, F(:, n-q_1:n))$, 则 $G_{D_{th}}(:, 1:q_1+1) = \text{fliplr}(r_{D_{th_1}}(p_1+1:\text{end}-p_1, 1:q_1+1))$, $G_{D_{th}}(:, n-q_1:m) = \text{fliplr}(r_{D_{th_2}}(p_1+1:\text{end}-p_1, q_1+1:\text{end}))$, $G_{D_{th}}$ 的其他元素为 0;

[0121] (2) 对 $G_{D_{tr}}$, 计算 $r_{D_{tr_1}} = \text{conv2}(k_{D_{tr_1}}, F(:, 1:q_1+1))$, $r_{D_{tr_2}} = \text{conv2}(k_{D_{tr_2}}, F(:, n-q_1:n))$, $G_{D_{tr}}(:, 1) = r_{D_{tr_1}}(p_1+1:\text{end}-p_1, q_1+1)$, $G_{D_{tr}}(:, n) = r_{D_{tr_2}}(p_1+1:\text{end}-p_1, q_1+1)$, $G_{D_{tr}}$ 的其他元素为 0;

[0122] (3) 对 $G_{D_{ht}}$, 计算 $r_{D_{ht_1}} = \text{conv2}(k_{D_{ht_1}}, F(1:p_1+1, :))$, $r_{D_{ht_2}} = \text{conv2}(k_{D_{ht_2}}, F(m-p_1:m, :))$, 则 $G_{D_{ht}}(1:p_1+1, :) = \text{flipud}(r_{D_{ht_1}}(1:p_1+1, q_1+1:\text{end}-q_1))$, $G_{D_{ht}}(m-p_1:m, :) = \text{flipud}(r_{D_{ht_2}}(p_1+1:\text{end}, q_1+1:\text{end}-q_1))$, $G_{D_{ht}}$ 的其他元素为 0;

[0123] (4) 对 $G_{D_{hh}}$ 有 $G_{D_{hh}} = G_{hh}$;

[0124] (5) 对 $G_{D_{rt}}$, 计算 $r_{D_{rt_1}} = \text{conv2}(k_{D_{rt_1}}, F(1:p_1+1, :))$, $r_{D_{rt_2}} = \text{conv2}(k_{D_{rt_2}}(1:p_1+1), F(m-p_1:m, :))$, 则 $G_{D_{rt}}(1, :) = r_{D_{rt_1}}(p_1+1, q_1+1:\text{end}-q_1)$, $G_{D_{rt}}(M, :) = \text{flipud}(r_{D_{rt_2}}(p_1+1, q_1+1:\text{end}-q_1))$, $G_{D_{rt}}$ 的其他元素为 0;

[0125] (6) 对 $G_{D_{lh}}$, 计算 $r_{D_{lh_1}} = \text{conv2}(k_{D_{lh_1}}, F(1:p_1+1, 1:q_1+1))$, $r_{D_{lh_2}} = \text{conv2}(k_{D_{lh_2}}, F(1:p_1+1, n-q_1:n))$, $r_{D_{lh_3}} = \text{conv2}(k_{D_{lh_3}}, F(m-p_1:m, 1:q_1+1))$, $r_{D_{lh_4}} = \text{conv2}(k_{D_{lh_4}}, F(m-q_1:m, n-q_1:n))$, 则 $G_{D_{lh}}(1:, 1:q_1+1) = \text{fliplr}(r_{D_{lh_1}}(p_1+1, 1:q_1+1))$, $G_{D_{lh}}(1:, n-q_1:n) = \text{fliplr}(r_{D_{lh_2}}(p_1+1, \text{end}-q_1:\text{end}))$, $G_{D_{lh}}(m, 1:q_1+1) = \text{fliplr}(r_{D_{lh_3}}(p_1+1, 1:q_1+1))$, $G_{D_{lh}}(m, n-q_1:n) = \text{fliplr}(r_{D_{lh_4}}(p_1+1, \text{end}-q_1:\text{end}))$, $G_{D_{lh}}$ 的其他元素为 0;

[0126] (7) 对于 $G_{D_{hl}}$, 计算 $r_{D_{hl_1}} = \text{conv2}(k_{D_{hl_1}}, F(1:p_1+1, 1:q_1+1))$, $r_{D_{hl_2}} = \text{conv2}(k_{D_{hl_2}}, F(1:p_1+1, n-q_1:n))$, $r_{D_{hl_3}} = \text{conv2}(k_{D_{hl_3}}, F(m-p_1:m, 1:q_1+1))$, $r_{D_{hl_4}} = \text{conv2}(k_{D_{hl_4}}, F(m-p_1:m, n-q_1:n))$, 则 $G_{D_{hl}}(1:p_1+1, 1) = \text{flipud}(r_{D_{hl_1}}(1:p_1+1, q_1+1))$, $G_{D_{hl}}(1:p_1+1, n) = \text{flipud}(r_{D_{hl_2}}(1:p_1+1, q_1+1))$, $G_{D_{hl}}(m-p_1:m, 1) = \text{flipud}(r_{D_{hl_3}}(\text{end}-p_1:\text{end}, q_1+1))$, $G_{D_{hl}}(m-p_1:m, n) = \text{flipud}(r_{D_{hl_4}}(\text{end}-p_1:\text{end}, q_1+1))$, $G_{D_{hl}}$ 的其他元素为 0;

[0127] (8) 对 $G_{D_{ll}}$, 计算 $r_{D_{ll_1}} = \text{conv2}(k_{D_{ll_1}}, F(1:p_1, 1:q_1))$, $r_{D_{ll_2}} = \text{conv2}(k_{D_{ll_2}}, F(1:p_1+1, n-q_1:n))$, $r_{D_{ll_3}} = \text{conv2}(k_{D_{ll_3}}, F(m-p_1:m, 1:q_1+1))$, $r_{D_{ll_4}} = \text{conv2}(k_{D_{ll_4}}, F(m-p_1:m, n-q_1:n))$, 则 $G_{D_{ll}}(1) = r_{D_{ll_1}}(p_1+1, q_1+1)$, $G_{D_{ll}}(1:p_1+1, n) = r_{D_{ll_2}}(p_1+1, q_1+1)$, $G_{D_{ll}}(m, 1) = r_{D_{ll_3}}(p_1+1, q_1+1)$, $G_{D_{ll}}(m, n) = r_{D_{ll_4}}(p_1+1, q_1+1)$, $G_{D_{ll}}$ 的其他元素为 0;

[0128] (9) 对 $G_{D_{pp}}$, 计算 $r_{D_{pp_1}} = \text{conv2}(\text{fliplr}(\text{flipud}(k_{D_{pp_1}})), F(1:p_1, 1:q_1))$, $r_{D_{pp_2}} = \text{conv2}(\text{flipud}(k_{D_{pp_2}}), F(1:p_1, n-q_1+1:n))$, $r_{D_{pp_3}} = \text{conv2}(\text{fliplr}(k_{D_{pp_3}}), F(m-p_1+1:m, 1:q_1))$, $r_{D_{pp_4}} = \text{conv2}(k_{D_{pp_4}}, F(1, 1))$, 则 $G_{D_{pp}}(1, 1) = r_{D_{pp_1}}(p_1, q_1)$, $G_{D_{pp}}(1, n) = r_{D_{pp_2}}(p_1, q_1)$, $G_{D_{pp}}(m, 1) = r_{D_{pp_3}}(p_1, q_1)$, $G_{D_{pp}}(m, n) = r_{D_{pp_4}}(p_1, q_1)$, $G_{D_{pp}}$ 的其他元素为 0。

[0129] 第 13 实施例

[0130] 记第 4 实施例的公式 $g_{D_{Ar_b}} = (K_{D_{tt}} - K_{D_{th}} + K_{D_{tr}} - K_{D_{ht}} + K_{D_{hh}} - K_{D_{hr}} + K_{D_{rt}} - K_{D_{rh}} + K_{D_{rr}})f$ 中, 除 $K_{D_{tt}}f$ 外各个模糊矩阵与图像矢量乘积的矩阵结果 (大小为 $m \times n$) 分别为 $G_{D_{th}}$ 、 $G_{D_{tr}}$ 、 $G_{D_{rt}}$ 、 $G_{D_{ht}}$ 、 $G_{D_{hh}}$ 、 $G_{D_{hr}}$ 、 $G_{D_{rh}}$ 和 $G_{D_{rr}}$, 本发明给出各个矩阵的计算方法, 具体为:

[0131] (1) $G_{D_{th}}$ 、 $G_{D_{tr}}$ 、 $G_{D_{rt}}$ 、 $G_{D_{ht}}$ 、 $G_{D_{hh}}$ 计算方法与第 12 实施例中同名矩阵的计算方法相同;

[0132] (2) 对 $G_{D_{hr}}$, 计算 $r_{D_{hr_1}} = \text{conv2}(k_{D_{hr_1}}, F(1:p_1+1, 1:q_1+1))$, $r_{D_{hr_2}} = \text{conv2}(k_{D_{hr_2}}$

$r_{d_{hr_2}}, F(1:p_1+1, n-q_1:n)$), $r_{d_{hr_3}} = \text{conv2}(k_{d_{hr_3}}, F(m-p_1:m, 1:q_1+1))$, $r_{d_{hr_4}} = \text{conv2}(k_{d_{hr_4}}, F(m-p_1:m, n-q_1:n))$, $G_{d_{hr}}(1:p_1+1, 1) = \text{flipud}(r_{d_{hr_1}}(1:p_1+1, q_1+1))$, $G_{d_{hr}}(1:p_1+1, n) = \text{flipud}(r_{d_{hr_2}}(1:p_1+1, q_1+1))$, $G_{d_{hr}}(m-p_1:m, 1) = \text{flipud}(r_{d_{hr_3}}(\text{end}-p_1:\text{end}, q_1+1))$, $G_{d_{hr}}(m-p_1:m, n) = \text{flipud}(r_{d_{hr_4}}(\text{end}-p_1:\text{end}, q_1+1))$, $G_{d_{hr}}$ 的其他元素为 0;

[0133] (3) 对 $G_{d_{rh}}$, 计算 $r_{d_{rh_1}} = \text{conv2}(k_{d_{rh_1}}, F(1:p_1+1, 1:q_1+1))$, $r_{d_{rh_2}} = \text{conv2}(k_{d_{rh_2}}, F(1:p_1+1, n-q_1:n))$, $r_{d_{rh_3}} = \text{conv2}(k_{d_{rh_3}}, F(m-p_1:m, 1:q_1+1))$, $r_{d_{rh_4}} = \text{conv2}(k_{d_{rh_4}}, F(m-p_1:m, n-q_1:n))$, 则 $G_{d_{rh}}(1, 1:q_1+1) = \text{fliplr}(r_{d_{rh_1}}(p_1+1, 1:q_1+1))$, $G_{d_{rh}}(1, n-q_1+1:n) = \text{fliplr}(r_{d_{rh_2}}(p_1+1, \text{end}-q_1:\text{end}))$, $G_{d_{rh}}(m, 1:q_1+1) = \text{fliplr}(r_{d_{rh_3}}(p_1+1, 1:q_1+1))$, $G_{d_{rh}}(m, n-q_1:n) = \text{fliplr}(r_{d_{rh_4}}(p_1+1, \text{end}-q_1:\text{end}))$, $G_{d_{rh}}$ 的其他元素为 0;

[0134] (4) 对 $G_{d_{rr}}$, 计算 $r_{d_{rr_1}} = \text{conv2}(k_{d_{rr_1}}, F(1:p_1+1, 1:q_1+1))$, $r_{d_{rr_2}} = \text{conv2}(k_{d_{rr_2}}, F(1:p_1+1, n-q_1:n))$, $r_{d_{rr_3}} = \text{conv2}(k_{d_{rr_3}}, F(m-p_1:m, 1:q_1+1))$, $r_{d_{rr_4}} = \text{conv2}(k_{d_{rr_4}}, F(m-p_1:m, n-q_1:n))$, 则 $G_{d_{rr}}(1, 1) = r_{d_{rr_1}}(p_1+1, q_1+1)$, $G_{d_{rr}}(1, n) = r_{d_{rr_2}}(p_1+1, q_1+1)$, $G_{d_{rr}}(m, 1) = r_{d_{rr_3}}(p_1+1, q_1+1)$, $G_{d_{rr}}(m, n) = r_{d_{rr_4}}(p_1+1, q_1+1)$, $G_{d_{rr}}$ 的其他元素为 0。

[0135] 第 14 实施例

[0136] 本发明给出第 10 和第 11 实施例中, 矩阵向量乘积 $K_{B_{tt}}f$ 的矩阵形式的计算方法, 具体为: $G_{B_{Zero}} = \text{conv2}(k, F)$ 。

[0137] 第 15 实施例

[0138] 本发明给出第 12 和第 13 实施例中, 矩阵向量乘积 $K_{D_{tt}}f$ 的矩阵形式的计算方法, 具体为: $G_{D_{Zero}} = \text{conv2}(k', F)$ 。

[0139] 第 16 实施例

[0140] 本发明给出 AR BCs 下, 当边角为 a 类型时, 滤波图像矩阵的计算方法, 具体为: $G_{B_{Ar_a}} = G_{B_{Zero}} - G_{B_{th}} + G_{B_{tr}} + G_{B_{rt}} - G_{B_{ht}} - G_{B_{hh}} + G_{B_{hl}} + G_{B_{lh}} - G_{B_{ll}} + G_{B_{pp}}$, 右式中各个矩阵的计算方法, 已由第 6、第 10 和第 14 实施例给出。

[0141] 第 17 实施例

[0142] 本发明给出 AR BCs 下, 当边角为 b 类型时, 滤波图像矩阵的计算方法, 具体为: $G_{B_{Ar_b}} = G_{B_{Zero}} - G_{B_{th}} + G_{B_{tr}} - G_{B_{ht}} + G_{B_{hh}} - G_{B_{hr}} + G_{B_{rt}} - G_{B_{rh}} + G_{B_{rr}}$, 右式中各个矩阵的计算方法, 已由第 7、第 11 和第 14 实施例给出。

[0143] 第 18 实施例

[0144] 本发明给出 AR BCs 下, 当边角为 a 类型时, 去卷积图像矩阵的计算方法, 具体为: $G_{D_{Ar_a}} = G_{D_{Zero}} - G_{D_{th}} + G_{D_{tr}} + G_{D_{rt}} - G_{D_{ht}} - G_{D_{hh}} + G_{D_{hl}} + G_{D_{lh}} - G_{D_{ll}} + G_{D_{pp}}$, 右式中各个矩阵的计算方法, 已由第 8、第 12 和第 15 实施例给出。

[0145] 第 19 实施例

[0146] 本发明给出 AR BCs 下, 当边角为 b 类型时, 滤波图像矩阵的计算方法, 具体为: $G_{D_{Ar_b}} = G_{D_{Zero}} - G_{D_{th}} + G_{D_{tr}} - G_{D_{ht}} + G_{D_{hh}} - G_{D_{hr}} + G_{D_{rt}} - G_{D_{rh}} + G_{D_{rr}}$, 右式中各个矩阵的计算方法, 已由第 9、第 13 和第 15 实施例给出。

[0147] 第 20 实施例

[0148] 本实例通过一个实验, 从正确性和时间消耗两个方面, 验证本发明所给出替代计

算方法的有效性。图像滤波计算中能得到真实图像,但在图像恢复中,真实图像常常不能直接获得,使得在图像恢复计算的开始计算,需用退化图像替代真实图像参与计算,因此本实验分别验证了本发明的替代计算方法在基于真实图像和退化图像时的有效性。实验在 PIV, 1.5GB 机器上采用 Matlab 完成。

[0149] 图 2 中为实验中采用的原始图像,是从标准图像 Cameraman.tif 生成的灰度图,大小为 256×256 ,表 1 为实验中采用的随机生成的点扩展函数,大小为 9×9 。

[0150] 图 3 为基于 Zeros BCs 的滤波图像。

[0151] 图 4 到图 11,分别为在不同边角类型时、不同参考图像(真实图像/同类型边界与边角下的退化图像)的乘积一与乘积二的替代计算结果(分别去掉计算结果中的最边的 1 行和 1 列,这部分像素异常)。可以发现,各个乘积一的边框处都过度自然,而各个乘积二尽管去掉最边的行和列,在边框处仍显黑暗,这是由于 AR BCs 的提出文献中,对图像边框像素依赖的设置方式所致。

[0152] 表 2 示出的是分别基于真实图像和退化图像,采用本发明的替代计算得到的乘积一和乘积二,与模糊矩阵与真实图像矢量乘积得到的结果(作为参照)间的平方根误差(rmse),以及直接将 Zero BCs 下的乘积一和乘积二,当作 AR BCs 下的乘积一和乘积二,所产生的 rmse。这里 rmse 的计算方法为 $rmse = \text{norm}(g_R - g_S) / \sqrt{\text{length}(g_R(:))}$,其中 g_R 由模糊矩阵与真实图像矢量间的乘积得到(计算 $H * f$, 或 $H' * f$), g_S 为采用本专利的替代计算结果,以及 Zero BCs 下的计算结果, $\text{norm}()$ 为 2-范数。可以看出:1) 当采用真实图像作为参考图像时,本发明替代计算方法的误差几乎为零;2) 在基于退化图像计算时,本发明计算误差比基于真实图像计算的误差略大,同样,这与 AR BCs 对边框像素的依赖方式有关。但无论是基于真实图像,还是基于退化图像,本发明替代计算的乘积一和乘积二的误差,都远远小于直接将 Zeros BCs 变换图像当作 AR BCs 变换图像(这种处理方式常被采用,因在本发明前, Zeros BCs 图像比 AR BCs 变换图像更易获得)。

[0153] 表 3 中示出的是直接用图像模糊矩阵与图像矢量乘积计算乘积一和乘积二,以及用本发明替代方法计算二者所消耗的时间。可以看出,用本发明替代计算的时间消耗,远远小于直接的模糊矩阵与图像矢量乘积计算的时间消耗。

[0154] 综合表 2 和表 3,可以看出,本发明的替代计算方法在正确性和时间消耗方面具有有效性。

[0155] 表 1

行号/列号	1	2	3	4	5	6	7	8	9
1	0.0244	0.0035	0.0252	0.0135	0.0015	0.0253	0.0105	0.0034	0.008
2	0.0133	0.0003	0.0149	0.0164	0.0154	0.0201	0.019	0.0053	0.0003
3	0.0225	0.0228	0.0108	0.0053	0.0013	0.0112	0.0068	0.0155	0.0098
4	0.0044	0.0051	0.0132	0.0097	0.0106	0.0127	0.0112	0.0161	0.0174
5	0.025	0.0076	0.0085	0.02	0.0078	0.0055	0.0238	0.0095	0.0024
6	0.0069	0.0169	0.0111	0.0174	0.0223	0.0164	0.0174	0.0147	0.0009
7	0.0064	0.0073	0.0058	0.0118	0.0004	0.0082	0.0054	0.0115	0.0156
8	0.0224	0.012	0.0148	0.0145	0.0196	0.0245	0.0214	0.0011	0.0155
9	0.0188	0.0017	0.0194	0.0203	0.0248	0.0186	0.0161	0.0007	0.0004

[0157] 表 2

边角条件	边界处理方法*	乘积一	乘积二
a 边角 类型	Real	3.97E-15	7.07E-15
	Blurred	0.3211	0.84912
	直接替换	8.6163	3.14E+01
b 边角 类型	R	3.72E-15	6.87E-15
	B	0.32117	0.90532
	直接替换	8.611	32.347

[0159] 注释:在表 2 的处理方法中, Real 指将真实图像作为替代计算中的参考图像 F, Blurred 指用模糊图像作为替代计算中的参考图像 F, 直接替换指直接将 Zero BCs 下的变换图像当作 ARBCs 下的变换图像。

[0160] 表 3

[0161]

边角条件	乘积类型	采用矩阵向量乘积	采用卷积替代计算
a 边角类型	乘积一	784.781	0.062
	乘积二	811.89	0.078
b 边角类型	乘积一	754.687	0.046
	乘积二	784.297	0.047

[0162] 注:在表 3 中各数值的单位为秒。

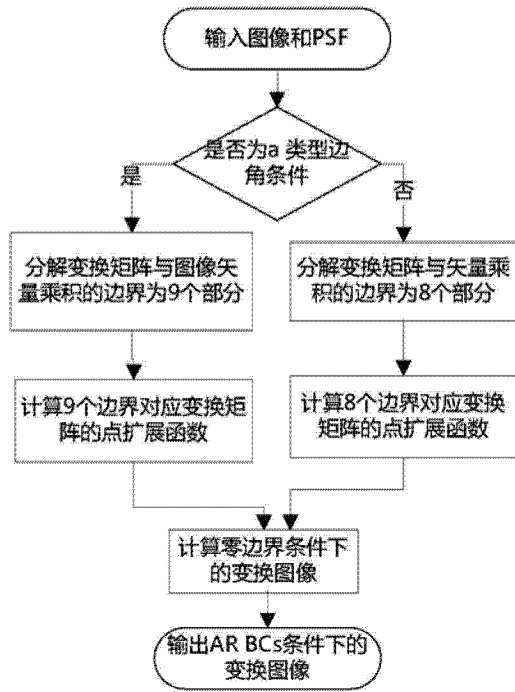


图 1

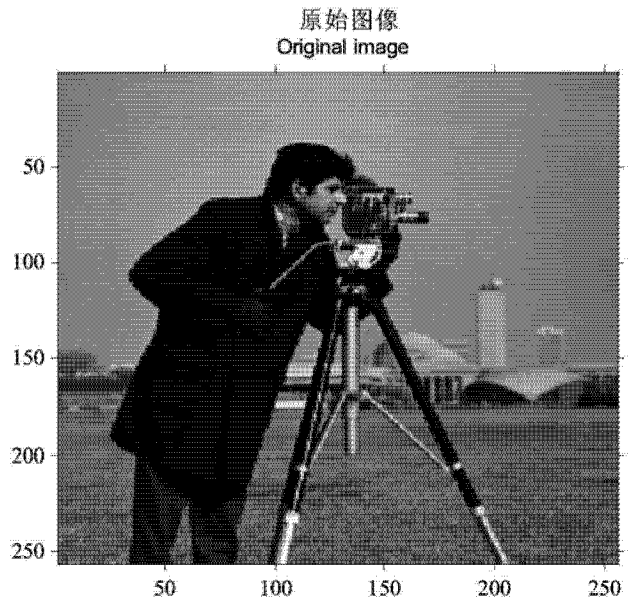


图 2

零边界条件下参考图像为原始图像的乘积—
blurred image under Zeros BCs

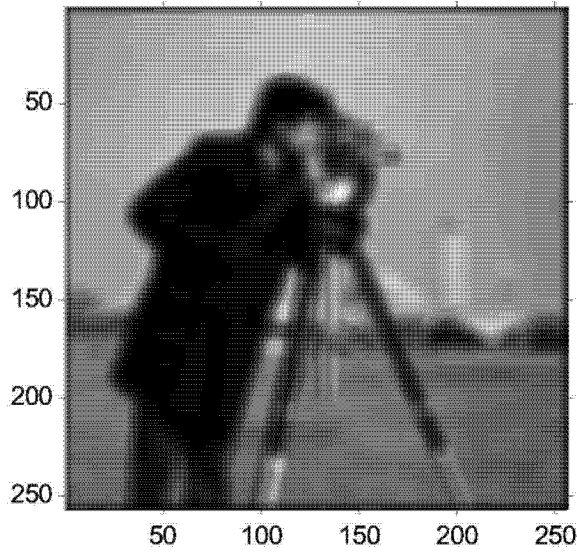


图 3

AR-a边界条件下参考图像为原始图像的乘积一
The first product image under AR-a from original image

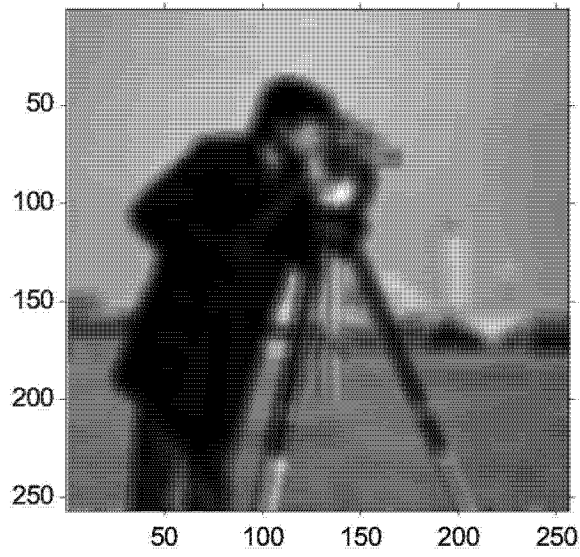


图 4

AR-a边界条件下参考图像为原始图像的乘积二
The second product image under AR-a from original image

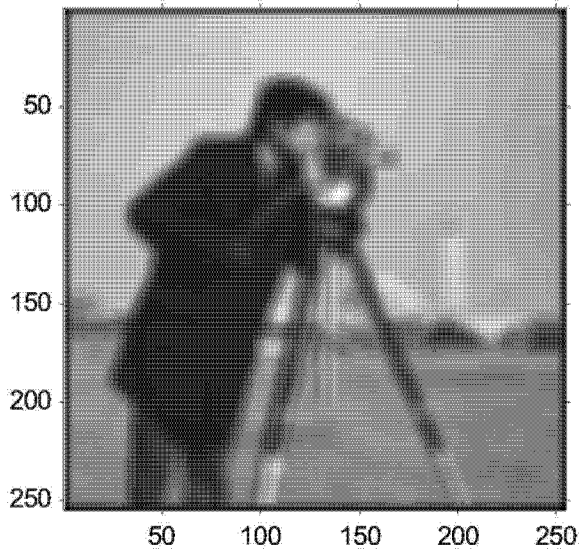


图 5

AR-b边界条件下参考图像为原始图像的乘积一
The first product image under AR-b form original image

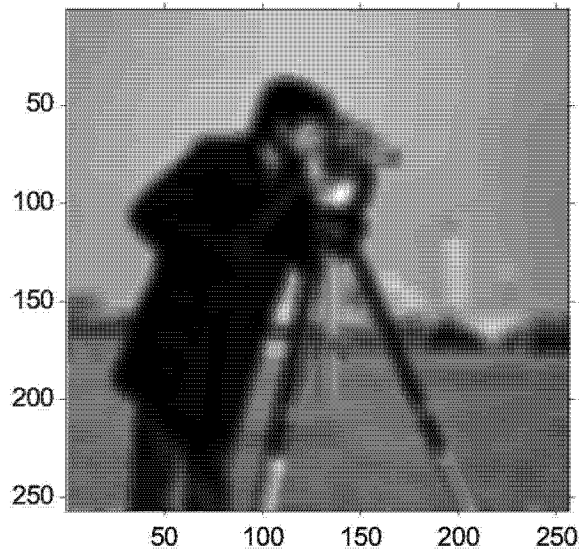


图 6

AR-b边界条件下参考图像为原始图像的乘积二
The second product image under AR-b from original image

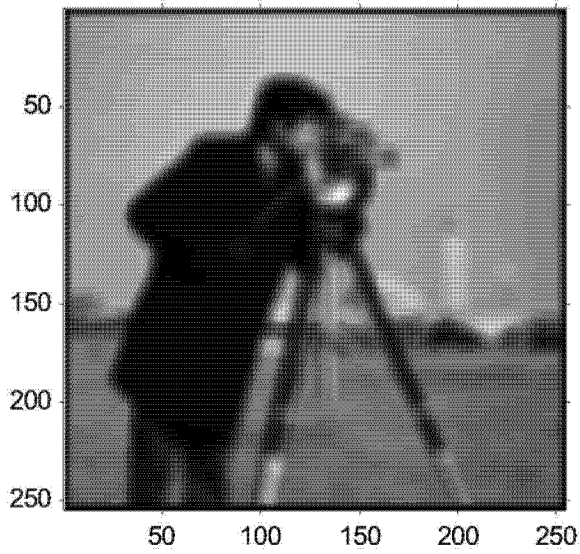


图 7

AR-a边界条件下参考图像为退化图像的乘积一
The first product image under AR-a from an AR-a blurred image

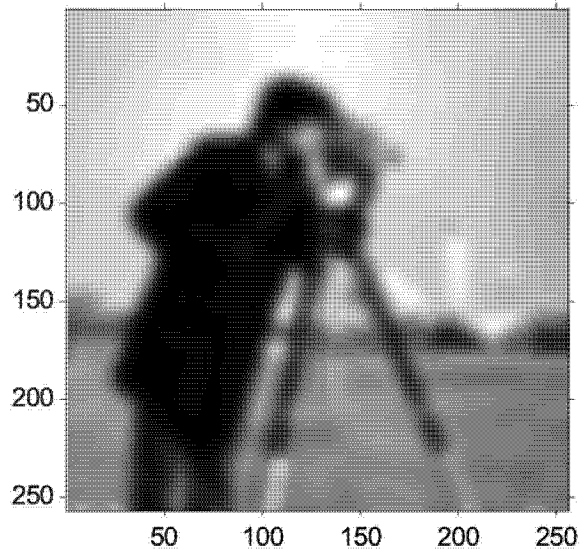


图 8

AR-a边界条件下参考图像为退化图像的乘积二
The second product image under AR-a from an AR-a blurred image

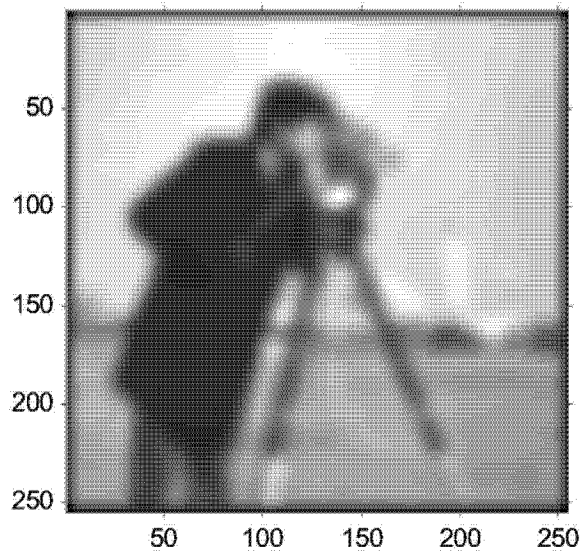


图 9

AR-b边界条件下参考图像为退化图像的乘积一
The first product image under AR-b from an AR-b blurred image

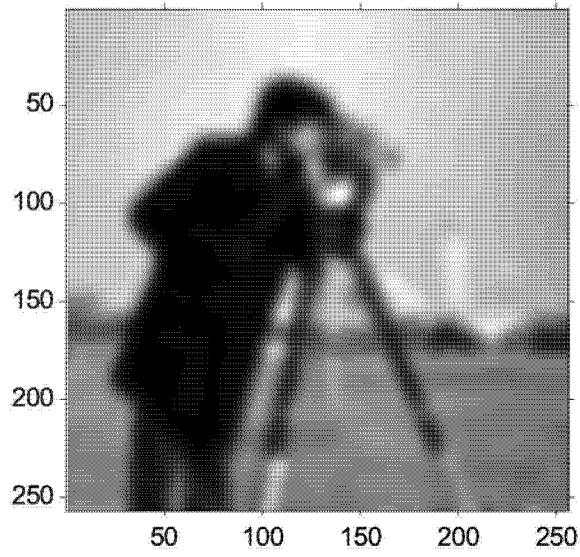


图 10

AR-b边界条件下参考图像为退化图像的乘积二
The second product image under AR-b from an AR-b blurred image

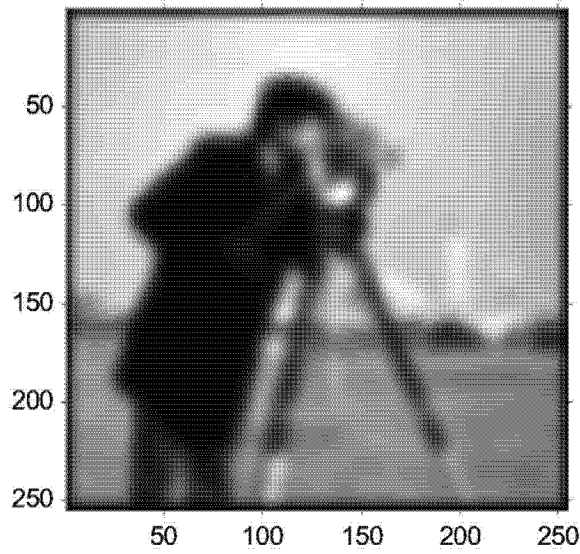


图 11