

基于统计特征的彩色图像快速插值方法

刘晓松, 杨 新, 汪 进

(上海交通大学图像处理与模式识别研究所, 上海 200030)

摘 要: 本文首先阐述了基于统计特征的图像插值方法, 该方法通过提取待插入像素所在区域的协方差矩阵和协方差向量, 得出适应于边缘位置和方向的插值权重. 为了把基于统计特征的图像插值方法应用于彩色图像插值领域, 本文提出了以下措施以提高计算速度: 仅对 Y 图像估计插值权重, 并同时应用到 R 、 G 、 B 三个分量的插值; 对边缘像素应用基于统计特征的图像插值方法, 而对非边缘像素应用简单的双线性插值, 即混合图像插值方法. 这些措施提高了计算速度, 并保证了图像质量. 实验表明了该算法在计算速度和插值图像质量方面的有效性.

关键词: 图像插值; 分辨率; 非线性滤波器; 颜色空间; 协方差特征

中图分类号: TN911.73 **文献标识码:** A **文章编号:** 0372-2112(2004)01-0029-05

Statistical Feature Based Fast Color Image Interpolation

LIU Xiaosong, YANG Xin, WANG Jin

(Institute of Image Processing and Pattern Recognition, Shanghai Jiaotong University, Shanghai 20030, China)

Abstract: This paper discusses a kind of statistical feature based image interpolation method at first. Using the covariance matrix and covariance vector in low resolution image, one can obtain the interpolation coefficients, which can adapt edge orientation. In order to employ statistical feature based interpolation in the area of color image interpolation, this paper present the following methods to increase the calculating speed: estimating the interpolation coefficients only based on Y image, and applying them to R , G , and B images interpolation simultaneously; For edge pixels, statistical feature based image interpolation method is exploited, while for non edge pixels, we only use simple bilinear interpolation. This method can be called mixture interpolation. Benefit from these measures, one can get high calculating speed while keeping the image quality unchanged. The experiments show that this method is both effective for calculating speed and interpolated image quality.

Key words: image interpolation; resolution; non linear filter; color space; covariance feature

1 引言

图像插值在高清电视、数字投影仪、图像打印驱动程序以及通常的图像的编辑和处理软件等方面有着广泛的应用, 因此也受到了很多学者的关注^[2~8]. 常见的图像插值方法有最近邻法(Nearest Neighbor)、双线性插值(Bilinear)法和双三次插值(Bicubic)法^[1], 然而由于边缘模糊效应(Edge Blurring)和边缘的阶梯形失真(Ringing Artifacts)等现象的存在, 有必要对其进行改进. 文献[12]提出了一种基于有理函数(Rational Function, 简称 RF)的非线性插值方法, 具有算法简单, 运算速度快的特点, 并能在一定程度上降低边缘失真效应. 降低边缘失真方面效果更好些的算法是自适应插值算法, 这类算法先提取出图像的边缘, 然后对边缘点进行“选择性”插值^[2,3]. 其缺点是在放大倍数较高的情况下对边缘的估计容易产生误差. 为了对这种算法进行改进, 一些学者提出了基于凸面投影集(Project Onto Convex Set 或 POCS)的插值算法^[4~6]. 这种方法

的主要思想是通过迭代运算对要求的高分辨率图像的边缘的位置和方向进行逼近, 在此基础上进行插值, 从而使图像边缘失真得到一定程度的抑制, 但是由于采用耗时的迭代运算, 速度很慢. 此外, 还有用小波和分形进行图像插值的研究^[7,8], 但在算法的速度和性能上仍有待提高.

随着数字技术的高速发展, 彩色图像的处理和显示设备在生活和工作中已必不可少. 有关彩色图像处理的研究已经成为一个非常活跃的研究领域^[9], 但是如何提高彩色图像的插值速度是需要仔细研究的课题.

文献[11]对应用于无损图像压缩的边缘自适应预测算法进行了研究. 应用这种算法, 首先要计算出待预测像素所在区域的协方差特征, 再根据协方差特征进行最小二乘法预测. 作者通过实验验证了该方法良好的边缘自适应特性. 本文把该插值算法推广到彩色图像插值领域, 并着重于提高彩色图像的插值速度. 实验结果表明, 经本文算法插值得到的图像不仅保留了普通基于协方差特征的图像插值算法的良好的边缘自

适应特性的特点,而且由于应用了快速算法,使计算速度甚至与基于有理函数的非线性插值算法相比也有所提高。

2 基于协方差特征的图像插值算法

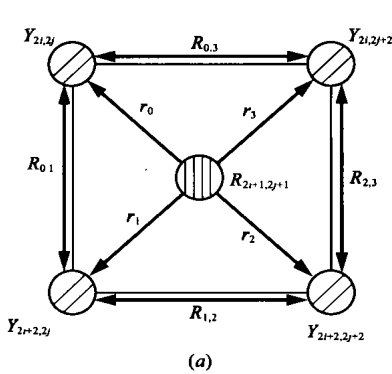
2.1 基于协方差特征的图像插值算法

假定有低分辨率图像 $X_{i,j}$, 插值后得到的高分辨率图像为 $Y_{k,l}$, 若放大倍数为 2, 有 $Y_{2i,2j} = X_{i,j}$. 现在要从四个低分

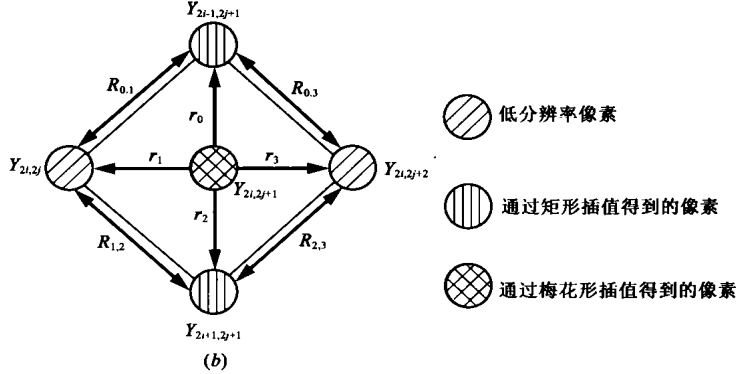
率像素 $\{Y_{2i,2j} = X_{i,j}, Y_{2i+2,2j} = X_{i+1,j}, Y_{2i,2j+2} = X_{i,j+1}, Y_{2i+2,2j+2} = X_{i+1,j+1}\}$ 估算出高分辨率像素 $Y_{2i+1,2j+1}$ (见图 1(a)), 有下式成立:

$$Y_{2i+1,2j+1} = \beta^T \cdot Y_{2i+1,2j+1} \quad (1)$$

其中: $\beta = (\beta_0, \beta_1, \beta_2, \beta_3)^T$; $Y_{2i+1,2j+1} = (Y_{2i,2j}, Y_{2i+2,2j}, Y_{2i,2j+2}, Y_{2i+2,2j+2})^T$.



(a) 矩形插值(Rect angular Interpolation)



(b) 梅花形插值(Quincunx Interpolation)

图 1 图像插值示意图

图 2 显示了放大倍数为 4 的情况下最近邻法和双线性方法的插值效果. 通过比较可以发现图 2(c) 的边缘阶梯形失真效应和图 2(d) 的边缘模糊效应, 同时图 2(c) 和图 2(d) 也分

别有不同程度的边缘模糊效应和边缘的阶梯形失真效应. 追其原因, 是由于普通的算法没有考虑在图像边缘附近插值的特殊性.

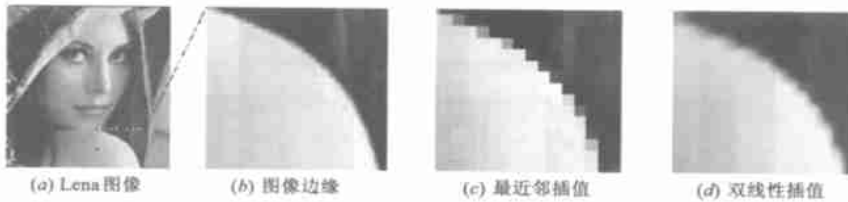


图 2 不同插值算法在图像边缘的表现(放大倍数为 4)

要尽可能的降低图像插值过程引入的两种边缘失真效应, 就要设法使式(1)中的插值权重 β_i 反映边缘的位置和方向. 这可以用图 3 来表示: 其中 P 代表待插值像素, N_n 为 P 的近邻低分辨率像素集合, 即图 1(a) 中的 $\{Y_{2i,2j}, Y_{2i+2,2j}, Y_{2i,2j+2}, Y_{2i+2,2j+2}\}$, 而 M_n 为用于计算插值权重 β_i 的包围 N_n 的低分辨率像素. 如果以均方差(Mean Square Error 或者 MSE)作为衡量估计误差的准则, 有下式成立:

$$MSE = \sum_{k \in M_n} \left\{ S_n - \sum_{l \in N_n} \beta_l \cdot S_l \right\}^2 \quad (2)$$

用矩阵形式表示, 有:

$$MSE = \|y - \beta \cdot C\|_2 \quad (3)$$

其中: $y = (S_{n-1}, S_{n-2}, \dots, S_{n-M})$, $S_{n-i} \in M_n$,

$$C = \begin{bmatrix} S_{n-1-1} & \dots & S_{n-1-N} \\ \vdots & \ddots & \vdots \\ S_{n-M-1} & \dots & S_{n-M-N} \end{bmatrix}$$

矩阵 C 的第 j 行的 N 个元素刚好为 S_{n-k} 的近邻域内的像素. 在满足式(3)中 MSE 最小的条件下, 可以求出:

$$\beta = (C^T \cdot C)^{-1} \cdot (C^T \cdot y) \quad (4)$$

这就是最小二乘法(Least Square, 或者 LS)估计. 如果设 $R_{XX} =$

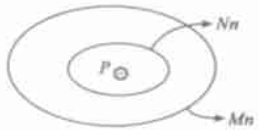


图 3 基于协方差特征的图像插值

$$(C^T C)/M, r_x = (C^T y)/M:$$

$$\beta = R_{XX}^{-1} \cdot r_x \quad (5)$$

实际式(5)是在图像符合平稳高斯过程的条件下基于协方差特征的估计^[11,12], 其中 R_{XX} 和 r_x 分别是通过低分辨率图像提供的信息对待插值像素所在区域的协方差矩阵和协方差向量的估计. 这样做的理由是: (1) 图像放大以后边缘的方向是不变的; (2) R_{XX} 和 r_x 提供了关于图像边缘位置和方向的信息.

对每个待插值像素计算 R_{XX} 和 r_x . R_{XX} 和 r_x 分别是 4×4 的矩阵和 4 维列向量:

$$R_{XX} = [R_{k,l}]_{4 \times 4}, r = (r_0, r_1, r_2, r_3)^T \quad (6)$$

R_{XX} 和 r_x 的计算方法为(以 r_1 的计算为例, 参考图 1(a), 其中 T 为窗口宽度)^[12]:

$$\hat{r}_1 = \frac{1}{(2T+1)^2} \sum_{k=-T}^{+T} \sum_{l=-T}^{+T} Y_{2k,2l} \cdot Y_{2k+2,2l-2} \quad (7)$$

计算完所有的 $Y_{2i+1,2j+1}$ 后, 再用类似的方法计算 $Y_{2i+1,2j}$ 和 $Y_{2i,2j+1}$ (图 1(b)).

2.2 几点说明

用 2.1 节中描述的算法我们可以得到放大倍数为 2^n 的

图像. 与双线性算法配合可以得到任意放大倍数的图像. 关于式(7)中窗口宽度 T 的选择, T 不能太小, 否则没有足够的信息计算出合适的 β_i ; T 也不能太大, 否则不仅运算量增大, 而且由于可能包含进去相关信息很少的点(窗口边缘附近的像素), 反而不能得到理想的 β_i . 实验表明 $T=3$ 比较合适. 现在再对算法的复杂性进行分析. 对于双线性插值, 只需要用 3 次加法和 1 次除法运算. 而本算法至少要用到 $14T+16$ 次乘法运算和 $14T+15$ 次加法运算, 还有一个 4×4 矩阵的求逆运算. 为了减少计算的复杂性, 可以采用下面的办法: 首先用简单的算法检测出低分辨率图像的边缘(比如 Sobel 算子), 然后对位于图像边缘的像素采用本文的插值算法, 而对非边缘像素使用简单的双线性插值. 这样做的理由是本文的算法对位于边缘位置的像素有效, 对非边缘像素效果并不明显. 而一般情况下一幅图像中位于边缘的像素只占整幅图像的很小一部分. 另外, 在式(5)中, R_{xx} 并不总是可逆的, 而这种情况一般出现在非边缘像素上, 运用上述混合插值方法就可以避免这种情况的出现.

3 快速彩色图像插值

我们已经讨论了基于统计特征的图像插值算法, 而且可以把该算法推广到彩色图像插值领域. 最简单的方法是直接对 R 、 G 、 B 三个颜色分量分别进行插值处理, 就可以得到比较理想的插值效果. 然而这样做计算量约为相应尺寸的灰度图像的三倍. 尽管应用 2.2 节叙述的混合插值方法可以有效地减少运算量, 但计算速度还是比普通的线性方法慢几倍, 而在很多应用场合算法的速度是至关重要的.

本文采用如下方法来提高彩色图像插值的速度, 见图 4. 首先从低分辨率彩色图像中提取出 Y 图像:

$$Y=0.301R+0.586G+0.113B \tag{8}$$

然后根据 Y 图像用 2.1 中叙述的算法计算出插值权重 β_i , 并对 R 、 G 、 B 图像同时应用 2.2 叙述的混合插值算法插值, 从而得到高分辨率的彩色图像. 用普通方法进行插值之所以比较耗时, 是由于它是分别对 R 、 G 、 B 三个颜色分量进行插值的. 这意味着对各个分量图像(R 图像 G 图像和 B 图像)都要分别应用第 2 节描述的算法, 即对每个分量图像的每个像素都要计算相应的插值权重 β_i , 这样计算量约为相应灰度图像插

值的 3 倍. 用图 4 所示的方法用低分辨率的 Y 图像提供的信息计算出插值权重 β_i 并应用于各个颜色分量图像的插值, 从而减少的运算量. 不考虑额外生成的 Y 分量所花的代价, 运算量将是普通方法的 1/3.

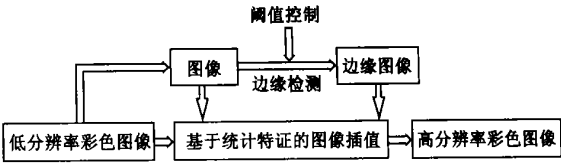


图 4 快速彩色图像插值

用 Y 分量代替各个颜色分量对方差矩阵和协方差向量进行估计的理由是: (1) 人眼对亮度分量的感觉比对颜色分量的感觉敏感; (2) Y 分量包含了 R 、 G 、 B 三个颜色分量的信息.

4 实验结果与分析

首先比较用 Bicubic 算法、基于有理函数的插值算法和本文算法插值得到的图像的质量. 用三个指标来评价图像质量: 峰值信噪比 (PSNR), 边缘峰值信噪比 (PESNR) 和均方差 (MSE). 如果用 I_s 表示测试图像, I'_s 表示插值图像, 则彩色图像 PSNR 和 PESNR 分别定义如下:

$$PSNR=10\times\log\left\{\frac{3\times M\times N\times 255^2}{\sum_s\sum_x\sum_y(I_s(x,y)-I'_s(x,y))^2}\right\},$$
$$PESNR=10\times\log\left\{\frac{3\times 255^2\times\sum_x\sum_ye(x,y)}{\sum_s\sum_x\sum_ye(x,y)\times(I_s(x,y)-I'_s(x,y))^2}\right\} \tag{9}$$

PSNR 反映了插值图像与原图像相符合的程度. 在计算 PESNR 的时候用到了边缘图像 $e(x,y)$, 其定义是: 如果 $e(x,y)=1$ 则表示像素 (x,y) 为边缘像素; 如果 $e(x,y)=0$ 表示非边缘像素. 可见 PESNR 反映了插值图像在边缘位置与原图像的相符合的程度. 有关 PESNR 的定义可以参阅参考文献[13].

表 1 对用三种插值法得到的图像的质量做了对比, 选用的 5 幅彩色图像的分辨率均为 512×512 . 使用的是 2.1 节中描述的算法, 即对三个颜色分量分别插值. 没有使用图 4 的快速算法. 分析表 1 中的数据, 可以看出 2.1 节中的算法是三种方法中图像质量最好的.

表 1 各种插值算法性能比较

放大倍数	图像	Bicubic			Rational Filter Based			Statistical Feature Based		
		PSNR	MSE	PESNR	PSNR	MSE	PESNR	PSNR	MSE	PESNR
2	Lena	29.03	81.27	22.49	31.36	47.50	25.04	32.14	39.77	26.10
	Baboon	21.07	508.28	18.68	22.25	387.17	19.75	22.60	357.33	20.11
	Peppers	28.97	82.44	21.64	31.05	51.09	24.02	31.59	45.11	24.95
	Airplane	28.39	94.30	21.43	31.15	49.85	23.92	31.52	45.84	24.36
	Birds	31.74	43.52	20.88	35.01	20.50	24.35	35.90	16.70	25.55
4	Lena	25.69	175.29	19.70	27.04	128.61	20.72	27.75	109.22	21.28
	Baboon	19.63	708.65	17.43	20.06	641.91	17.73	20.22	618.18	17.84
	Peppers	25.41	187.23	18.88	26.92	132.28	19.93	27.64	112.07	20.45
	Airplane	24.44	234.17	18.39	25.83	169.86	19.33	26.29	152.83	19.65
	Birds	27.52	115.23	17.34	29.27	76.87	18.64	30.29	60.83	19.59

下面对各种算法的计算速度进行验证. 为了方便叙述我们对各种算法的名称做个约定. 首先是对 3 个颜色分量的每个像素分别运用本文的插值方法, 即不使用 2.2 节的混合插值方法和图 4 的快速算法, 称为方法 1; 使用图 4 的快速算法, 但不使用 2.2 节的混合插值方法, 即不进行边缘检测而对每个像素都用 2.1 的算法, 称为方法 2; 使用图 4 的快速算法并进行边缘检测, 称为方法 3. 其中方法 1 和方法 2 由于没有使用边缘检测, 所以只要图像的大小和放大倍数确定了计算速度就大致确定了. 表 2 对方法 1 和方法 2 的插值速度和质量做了比较.

表 2 两种插值方法速度和性能比较

图 像	插值方法	计算时间(s)	PSNR	PESNR	MSE
Lena(128×128) 放大 4 倍	方法 1	15.38	27.75	21.28	109.22
	方法 2	5.26	27.67	21.26	111.20
Peppers(128×128) 放大 4 倍	方法 1	15.34	27.64	20.45	112.07
	方法 2	5.24	27.46	20.40	116.82

我们的实验是在 P III600, 128M 内存的计算机上进行的, 编译器为 VC6. 分析实验数据可以发现用这两种方法插值得到的图像的质量几乎完全一样, 然而方法 2 的计算速度却是方法 1 的 3 倍, 从而验证了本文第三部分的结论. 为了进一步提高速度, 使用了图 4 的快速插值算法, 边缘检测算子为 Sobel 算子. 图 5 比较了在不同的阈值下 Peppers 图像和 Lena 图像在放大 4 倍的情况下 PSNR 和计算速度的比较.

表 3 方法 2 与其他插值算法速度比较

图像	放大倍数	Bilinear (s)	Bicubic (s)	RF Based (s)	方法 2(s)
Lena(128×128)	4	0.44	0.57	1.43	5.26
Peppers (128×128)	4	0.43	0.56	1.41	5.24

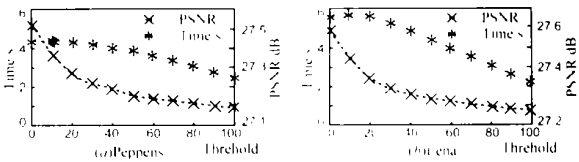


图 5 不同阈值下 PSNR 与计算速度比较

阈值为 0 的时候就是上述方法 2. 当阈值取 0 到 60 时, 计算速度显著提高, 再继续增大阈值计算速度提高并不明显, 而在整个过程中 PSNR 变化很小. 以图 5(a) 为例, 阈值为 0 时耗时 5.24s, PSNR 为 27.46; 而阈值取 60 时耗时 1.36s, PSNR 为 27.40, 计算时间减少为原来的 25.8%, 而 PSNR 只降低了 0.22%, 从而证明了图 4 快速算法的有效性. 表 4 为阈值取 60 时方法 3 与其他算法的比较.

表 4 方法 3 与其他插值算法比较

图 像	插值方法	计算时间(s)	PSNR	PESNR	MSE
Lena(128×128) 放大 4 倍	Bicubic	0.53	25.69	19.70	175.29
	RF	1.38	27.09	20.77	127.04
	方法 3(Th=60)	1.26	27.53	21.17	114.79
	方法 3(Th=100)	0.79	27.39	21.06	118.63
Peppers(128×128) 放大 4 倍	Bicubic	0.53	25.41	18.88	187.23
	RF	1.39	26.99	19.99	130.07
	方法 3(Th=60)	1.36	27.40	20.63	118.41
	方法 3(Th=100)	0.88	27.30	20.31	121.17

由试验数据可见, 方法 3 在速度和图像质量上都比基于有理函数的插值算法更有优势. 图 6(e) 和图 6(f) 分别是阈值为 50 和 100 时 Flower 图像放大 4 倍效果.

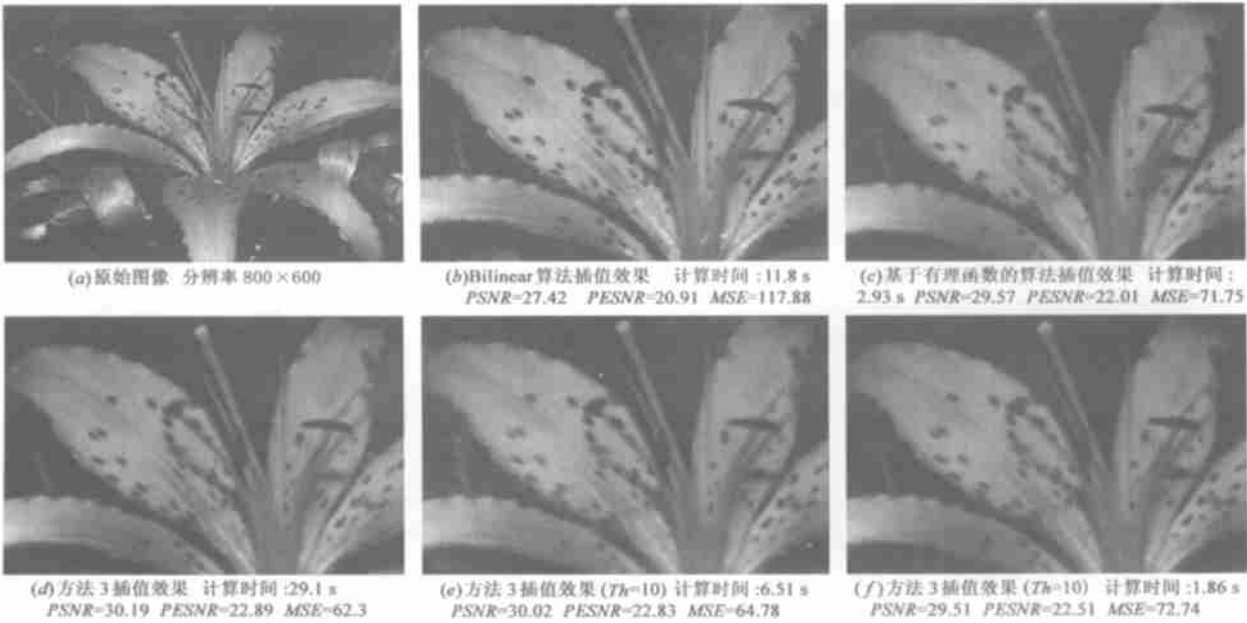


图 6 几种插值算法效果图(放大倍数为 4)

5 总结

本文首先阐述了基于统计特征的图像插值方法, 并证明了其良好的边缘自适应特性. 在此基础上提出了基于统计特征的快速彩色图像插值算法: 首先提取 Y 图像并检测其边缘, 对非边缘像素使用双线性插值; 对非边缘像素, 用基于统计特征的图像插值方法计算插值权重, 并同时应用于 RGB 三个分量图像的插值. 该算法大大提高了彩色图像插值速度, 而保持图像的质量基本不变. 实验证明该算法的有效性.

参考文献:

- [1] A K Jain. Fundamentals of Digital Image Processing [M]. Prentice Hall, 1988.
- [2] S Carrato, G Ramponi, S Marsi. A simple edge sensitive image interpolation filter [J]. Proceeding of ICIP 1996: 711– 714.
- [3] K Jensen, D Anastassiou. Subpixel edge localization and the interpolation of still images [J]. IEEE trans. Image processing, 1995, 4(3): 285 – 295.
- [4] J P Allebach, W P Wong. Edge directed interpolation [J]. Proceeding of ICIP, 1996, 3: 701– 711.
- [5] K Ratakonda, N Ahuja. POCS based adaptive image magnification [J]. Proceeding of ICIP 1998: 203– 207.
- [6] L S DeBrunner, V DeBrunner, M Yao. Edge retaining asymptotic projections onto convex sets for image interpolation [A]. Image Analysis And Interpretation, Proceedings of 4th IEEE Southwest Symposium [C]. Austin, TX, 2000. 78– 82.
- [7] S G Chang, Z Cvetkovic, M Vetterli. Resolution enhancement of images using Wavelet transform extrema extrapolation [J]. Proceeding of ICASSP, 1995, 4: 2379– 2382.
- [8] M Gharavi Alkhansari. Resolution enhancement of images using a fractal model [D]. Ph. D. thesis, university of Illinois at Urbana, 1996.
- [9] G Shama, H J Trussell. Digital color imaging [J]. IEEE trans. Image processing, 1997, 6(7): 901– 932.

- [10] Cheikh F A, Khiji L, Gabbouj M, Ramponi G. Color image interpolation using vector rational filters [A]. Proceedings of the SPIE' 98, IS&T/SPIE Conference on nonlinear image processing IV [C]. San Jose, California, USA, 1998. 242– 249.
- [11] Xin Li, M T Orchard. Edge directed prediction for lossless compression of natural images [J]. IEEE Trans. Image Processing, 2001, 10(6): 813 – 817.
- [12] G Ramponi. Image processing using rational functions [A]. Proceedings of the Cost 254 Workshop [C]. Budapest, Hungary, 1997. 6– 7.
- [13] J Mukherjee, R Parthasarathi, S Goyal. Markov random field processing for color demosaicing [J]. Pattern Recognition Letters, 2001, 22: 339– 351.

作者简介:



刘晓松 男, 1975 年生于辽宁绥中, 1998 年毕业于西安交通大学电子工程系, 2003 年 3 月获得上海交通大学工学硕士学位, 主要研究兴趣为: 模式识别、图像插值以及彩色图像处理等, 现就职于美国国家仪器公司上海研发中心, 从事数字信号处理相关算法的研究.



杨 新 男, 1950 年 11 月生于上海, 1982 年 12 月在西北工业大学取得工学硕士学位, 1995 年在比利时布鲁塞尔自由大学电子系获应用科学博士学位, 1995 年至 1997 年在西北工业大学计算机系进行博士后研究, 现为上海交通大学图像处理与模式识别研究所教授, 博士生导师, 主要从事图像中偏微分方程的研究, 运动目标检测, 图像匹配, 三维物体可视化等研究, 已在国内外学术刊物和会议上发表文章 60 余篇.