

文章编号:1001-9081(2006)12-2880-04

基于边缘最大相关性的快速图像插值算法

党向盈^{1,2}, 吴锡生¹, 赵 勇²

(1. 江南大学 信息工程学院, 江苏 无锡 214122; 2. 徐州教育学院 信息技术系, 江苏 徐州 221006)

(dxy8088@163.com)

摘 要: 处理视频、网络信号要求有较好的视觉质量和较低的算法复杂度、运算量。传统的图像插值算法会产生细节模糊和边缘锯齿化, 为此, 提出了一种基于图像边缘多方向最大相关性的快速数字图像插值算法。为了得到高分辨率图像相关信息, 根据图像空间域内多个方向邻近像素点的相关特性, 利用与待内插值点相邻 6 个降采样像素值。实验证明, 该算法降低了运算复杂度, 有效地保持了边缘信息, 得到了视觉质量较好的插值图像。

关键词: 图像插值; 最大相关性; 边缘保持

中图分类号: TP391.41 **文献标识码:** A

Fast image interpolation algorithm based on edge-directed max-relativity

DANG Xiang-ying^{1,2}, WU Xi-sheng¹, ZHAO Yong²

(1. School of Information Engineering, Southern Yangtze University, Wuxi Jiangsu 214122, China;

2. Department of Information Technology, Xuzhou Education College, Xuzhou Jiangsu 221006, China)

Abstract: Better subjective quality and lower computational complexity of the algorithm are important in video and network signal processing. And the smooth effect and staircase of image details or edges are inevitable during conventional image interpolation. A new rapid image interpolation algorithm based on edge-directed max-relativity was proposed. The algorithm took account of the multiple directions relativity of local neighbors and made full use of six down-sampled neighbors involved in the low-resolution image to estimate relative information at high resolution. Experimental results demonstrate that the algorithm not only decreases the computational complexity but also substantially improves the visual quality of the interpolated images.

Key words: image interpolation; max-relativity; edge-preserving

0 引言

图像插值算法经常被用于图像扩大和局部图像缩放, 其技术广泛用于静态图像和视频信号压缩解码显示。经典的图像内插方法有邻近插值法^[1]、双线性插值法^[1]和双立方插值法^[1,2]等。这些方法算法简单, 容易实现, 但是经过插值后的图像边缘模糊或之字形比较明显。文献[3]提出了一种局部自适应数字图像放大插值算法, 文献[4]给出了一种新的边缘方向插值算法, 文献[5]提出了一种最大相关性自适应数字图像插值算法。以上这些算法明显地改善了图像边缘的模糊效应, 但都存在算法复杂、运算速度慢的缺点。因为数字图像、录影序列等本身数据信息量就很大, 在网络传输、多媒体通信过程中, 网络带宽有限和硬件设备的制约, 使这些算法达不到图像快速插值的目的。文献[6]提出了一种针对边缘自适应的图像快速插值算法。该算法成功地实现了从 QCIF 录影序列到 CIF 序列的实时扩大, 在低比特率情况下得到了较好的图像质量。但该算法仍有一些不足, 在高分辨率图像中,

它通过间接利用与其相邻的新生成的像素点来估计待插入像素点的值, 这样会产生逐步误差累加和边缘方向误判断, 最后造成比较大的误差值。本文充分挖掘图像边缘局部结构特征, 直接利用与待内插像素点相邻的 6 个降采样点的信息和多方向最大相关等特征, 确定待插入点的值, 这样可以降低算法复杂度和运算量, 减少误差累加, 从而有效地保持图像的边缘信息, 加快插值速度。

1 边缘最大相关性快速图像插值算法

设有一幅 $H \times V$ 的低分辨率图像 $X_{i,j}$, 拟插值后得到的 $2H \times 2V$ 高分辨率图像为 $Y_{i,j}$, 那么 $Y_{2i,2j} = X_{i,j}$ 。常用图像插值算法和文献[4~7]算法的本质是利用直接邻接点的灰度值, 或邻接点间灰度值变化率, 或局部结构特性对待采样点进行内插。而本文算法首先根据像素点局部特征, 通过预先设置的阈值把原像素点分为边缘点和非边缘点两类, 然后对两类像素点采用不同的插值算法。具体处理方法如图 1 所示。

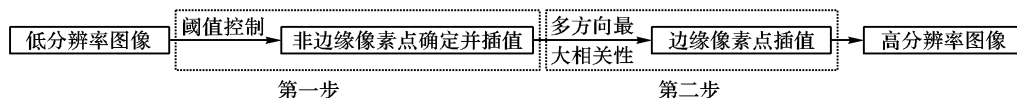


图 1 本文图像插值算法处理流程

收稿日期:2006-06-05; 修订日期:2006-08-23

作者简介: 党向盈(1978-), 女, 江苏徐州人, 硕士研究生, 主要研究方向: 人工智能、图像处理与模式识别; 吴锡生(1959-), 男, 江苏无锡人, 副教授, 博士, 主要研究方向: 人工智能、模式识别、图像处理; 赵勇(1966-), 男, 江苏徐州人, 讲师, 硕士, 主要研究方向: 多媒体技术、图像处理。

1.1 阈值控制的非边缘像素点插值

在图 1 所示的第一步中,首先选择局部 3×3 窗口(如图 2(a)所示),这样能有效减少计算装载量和算法的结构复杂性。然后,分别沿水平、垂直和两个对角共四个方向计算像素间差值,再与预置阈值比较。如果像素点间差值比预置阈值小,则待内插像素点归类为非边缘像素点,直接使用双线性插值算法;反之,如果像素点间差值比预置阈值大,则待内插像素点就归为边缘像素点。该类像素点使用本文算法插值。

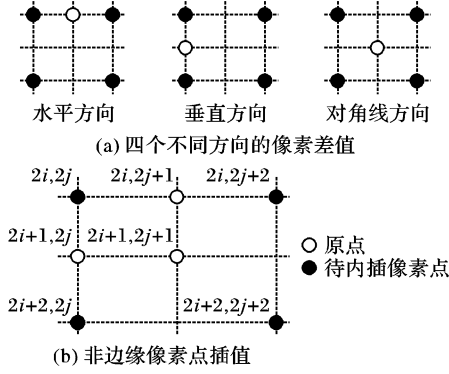


图 2 阈值控制像素点分类插值

利用阈值控制方式,判断原像素点所属类别算法描述如下(参见图 2(b)):

$$T1 = |Y_{2i,2j} - Y_{2i+2p,2j+2q}|, p, q \in (0,1), (1,0)$$

$$T2 = |Y_{2i,2j+2} - Y_{2i+2,2j}|$$

$$T3 = |Y_{2i,2j} - Y_{2i+2,2j+2}|$$

if $T1 < threshold$ then

$$Y_{2i+p,2j+q} = (Y_{2i,2j} + Y_{2i+2p,2j+2q})/2$$

else $Y_{2i+p,2j+q}$ 为边缘像素点

end

if $T2 < threshold$ and $T2 < threshold$ then

$$Tmin = \min(T2, T3)$$

if $Tmin == Tk (k \in \{2,3\})$ then

$$\text{线性插值 } Y_{2i+1,2j+1}$$

end

else if $Tk < threshold$ then

$$\text{线性插值 } Y_{2i+1,2j+1}$$

end

end

1.2 基于边缘多方向最大相关性的插值算法

由于人眼对图像边缘特别敏感,插值图像边缘部分的分辨率对一幅图像的质量有非常重要的影响。必须考虑图像边缘在多个方向上的信息,在图像中,图像边缘信息体现为方向上像素值的相似性及相关性,即沿着边缘方向像素值比较接近。

在图 1 所示的第二步中,为了估计边缘像素点 $Y_{i,j}$,把 $Y_{i,j}$ 分为两类:位于奇数行偶数列,或位于偶数行奇数列的像素点 $Y_{i,j} (i+j \text{ 为奇数})$ 定义为 I 类;位于奇数行奇数列的像素点 $Y_{i,j} (i+j \text{ 为偶数})$ 定义为 II 类。

这两类像素点插值又分两步进行处理。

步骤 1:先对 I 类像素点的处理。

这里以 $Y_{i,j}$ 位于奇数行偶数列的像素点为例进行说明(如图 3(b))。

文献[4~7]插值算法的基本思想都是通过原像素点首先插值位于行列和为偶数的像素点 $Y_{i,j} (i+j \text{ 为偶数})$,然后像素 $Y_{i,j} (i+j \text{ 为偶数})$ 再结合原像素点插值行列和为奇数的像

素点 $Y_{i,j} (i+j \text{ 为奇数})$ 。本文算法正好相反,通过分析像素点的空间域的局部特征信息,提出先插值行列和为奇数的像素点 $Y_{i,j} (i+j \text{ 为奇数})$,再结合原像素点,插值行列和为偶数的像素点 $Y_{i,j} (i+j \text{ 为偶数})$,这样在保持较好视觉质量的同时,具有较低的算法复杂度。

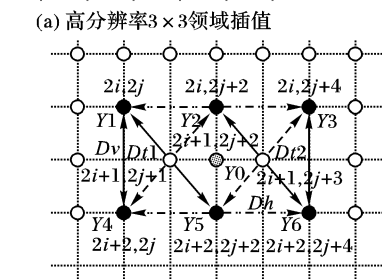
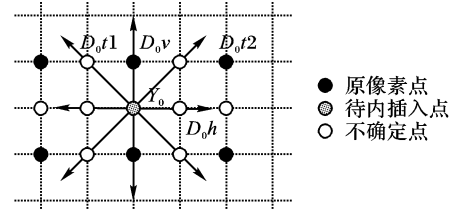


图 3 位于奇数行偶数列像素点多方向最大相关性插值

设待内插像素点为 $Y0$ 。计算像素点 $Y0$ 时,若通过计算高分辨率图像空间 3×3 领域内四个方向的相关性 $D0k (k \in \{h, v, t1, t2\})$,分别设四个方向相关性:水平方向为 $D0h$ 、垂直为 $D0v$ 、两个斜方向分别为 $D0t1, D0t2$,如图 3(a) 所示。估计四个方向的最大相关性,并在该最大相关性方向上插值。因为四个方向中三个方向上的像素值都具有不确定性,这对 $Y0$ 像素值及其边缘方向的确定会产生误差。

为了降低算法复杂度和减少误差累加,考虑在边缘同一方向,像素间具有相似和相关特性,本文直接利用 $Y0$ 邻近的 6 个低分辨率空间像素点所组成的 3×2 领域,计算在 3×2 领域内各方向的相关性 $Dk (k \in \{h, v, t1, t2\})$ 来近似估计高分辨率图像 3×3 领域各方向相关性 $D0k$ 。

如图 3(b) 所示,设 $Y1 \sim Y6$ 为图像中 6 个降采样像素点, $Y0$ 为待内插像素点,分别设四个方向相关性:水平为 Dh 、垂直方向为 Dv 、两个斜方向为 $Dt1, Dt2$ 。计算像素间的相关性 $Dk (k \in \{h, v, t1, t2\})$,如公式(1)所示:

设 $f(k)$ 表示的各点的像素值($k \in \{Y0, Y1, Y2, Y3, Y4, Y5, Y6\}$):

$$Dv = |f(Y1) - f(Y4)| + |f(Y3) - f(Y6)|$$

$$Dh = |f(Y1) - f(Y3)| + |f(Y4) - f(Y6)|$$

$$Dt1 = |f(Y1) - f(Y5)| + |f(Y2) - f(Y6)| \quad (1)$$

$$Dt2 = |f(Y2) - f(Y4)| + |f(Y3) - f(Y5)|$$

$$Dmin = \min(Dk), k \in \{h, v, t1, t2\}$$

$Dmin$ 意味着像素点在该方向有最高的相关性,即确定出该点的边缘方向,待内插像素点沿最小差异 $Dmin$ 方向内插。

算法思想描述如下:

1) 如果 $Dmin == Dv$,则 $f(Y0) = (f(Y2) + f(Y5))/2$ 。

2) 如果 $Dmin == Dh$,因为 $Y0$ 沿水平方向邻近像素点 $Y_{2i+1,2j+1}$ 和 $Y_{2i+1,2j+3}$ 的值不确定,所以 $Y0$ 的插值又有下列三种情况:

(a) 若 $Y_{2i+1,2j+1}$ 和 $Y_{2i+1,2j+3}$ 都已知,则 $f(Y0) = (f(2i+1,$

$$2j+1) + f(2i+1, 2j+3))/2;$$

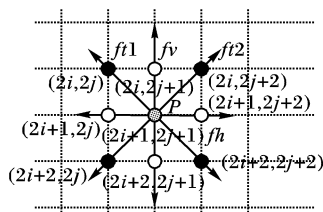
(b) 若 $Y_{2i+1, 2j+1}$ 和 $Y_{2i+1, 2j+3}$ 有一个点已知, 则 Y_0 的值就被赋值该已知点的值;

(c) 若 $Y_{2i+1, 2j+1}$ 和 $Y_{2i+1, 2j+3}$ 都未知, 则 $f(Y_0) = (f(Y_1) + f(Y_3) + f(Y_4) + f(Y_6))/4$ 。

3) 如果 $D_{\min} = Dt1$, Y_0 的像素值若沿斜方向上的不确定像素点 $Y_{2i, 2j+1}$ 和 $Y_{2i+2, 2j+3}$ 而得, 那么因为 $Y_{2i, 2j+1}$ 和 $Y_{2i+2, 2j+3}$ 的插值也是由邻近像素点所得, 存在一定的误差, 为了减少运算复杂性, 直接由 Y_1 和 Y_6 确定, 即 $f(Y_0) = (f(Y_1) + f(Y_6))/2$ 。

4) 如果 $D_{\min} = Dt2$, 同 3): $f(Y_0) = (f(Y_3) + f(Y_4))/2$ 。

位于偶数行奇数列的像素点 $Y_{i,j}$, 可用与上面类似的方法, 通过邻近 6 个低分辨率像素点所成的 2×3 领域得到其待内插值。



● 原像素点 ○ 待内插点 ○ 不确定点
图 4 位于偶数行偶数列像素点插值

步骤 2: 对 II 类像素的处理。

待内插点 P (设 P 为 $Y_{i,j}$, 且 $i+j$ = 偶数) 通过高分辨率 3×3 领域内四个方向的邻近像素点自适应插值生成。它的邻近点由原像素点和 I 类中已生成的新插值的 $Y_{i,j}$ ($i+j$ = 奇数) 两部分组成, 并且这些邻近点都有了确定的值。

设 P 的四个方向像素间差值分别设为: 水平方向 fh , 垂直方向 fv , 两个斜方向 $ft1, ft2$ 。如图 4 所示。比较各个方向像素差值 fk 之间的关系 ($k \in \{h, v, t1, t2\}$), 如公式 (2) 所示。

$$\begin{aligned} fh &= |f(2i+1, 2j) - f(2i+1, 2j+2)| \\ fv &= |f(2i, 2j+1) - f(2i+2, 2j+1)| \\ ft1 &= |f(2i, 2j) - f(2i+2, 2j+2)| \\ ft2 &= |f(2i, 2j+2) - f(2i+2, 2j)| \\ fmin &= \min(fk), k \in \{h, v, t1, t2\} \end{aligned} \quad (2)$$

最小的差值 $fmin$ 的确定, 预示着边缘像素点 P 在此方向最相关, 即为 P 的边缘方向。

例如, 若 $fmin = ft1$ 时, 与 P 点最密切相关的像素点是 $Y_{2i, 2j}$ 和 $Y_{2i+2, 2j+2}$, 那么 $f(2i+1, 2j+1) = (f(2i, 2j) + f(2i+2, 2j+2))/2$ 。

位于整个图像四周边界上的像素点, 无法用本文算法插值, 则直接采用线性算法插值。

对于彩色图像, 一般情况下都使用 RGB 模式, R 对应红色, G 对应绿色, B 对应蓝色, 它们统称为三基色, 这三种色彩的不同搭配, 就可以搭配成各种现实中的色彩, 此时彩色图像都需用 3 个样本组成的一个三维数组表示, 其中每个样本用于表示该像素的一个基本颜色。据此, 本文提出了彩色图像的算法。对彩色图像 R, G, B 三个分量分别按以上方法进行插值, 插值后的 R, G, B 三个分量再组合成彩色图像。

3 实验结果和分析

用本文提出的算法对灰度图像和彩色图像进行实际的插值, 并与文献 [6] 中的算法从客观质量和主观质量两方面进行比较分析原图像和插值图像。

客观质量通过三个评价标准进行评判: 均方差 (MSE), 峰值信噪比 (PSNR) 和边缘峰值信噪比 (PESNR)。它们的定义如公式 (3) 所示。PSNR 反映了插值图像与原图像相符合的程度。在计算 PESNR^[7,8] 的时候用到了边缘图像 $e(i, j)$, 其定义是: 如 $e(i, j) = 1$ 则表示 $Y_{i,j}$ 为边缘像素; 如果 $e(i, j) = 0$ 表示 $Y_{i,j}$ 为非边缘像素。可见 PESNR 反映了插值图像在边缘位置与原图像相符合的程度。一般情况下, PSNR、PESNR 愈大, 说明所恢复图像质量愈高。

$$MSE = \sum_{i=1}^N \sum_{j=1}^M (f(i, j) - g(i, j))^2$$

$$PSNR = 10 \log_{10} \frac{L^2}{MSE}$$

$$PESNR = 10 \log_{10} \frac{L^2 \times \sum_{i=1}^N \sum_{j=1}^M e(i, j)}{\sum_{i=1}^N \sum_{j=1}^M e(i, j) \times (f(i, j) - g(i, j))^2} \quad (3)$$

说明: 其中 M, N 分别是图像行 (i)、列 (j) 像素点的个数; $f(i, j), g(i, j)$ 分别对应原图像和插值图像在点 $Y_{i,j}$ 上的取值; L 是图像中灰度值的取值范围, 对 8 比特的灰度图像而言 $L = 255$ 。

表 1 显示了放大 2 倍和放大 4 倍的四幅灰度插值图像的客观性能比较。从这两组实验可以看出, 本文算法的三种客观评价性能指标整体优于文献 [6]。表 2 是插值放大 2 倍的彩色图像 PSNR 各分量的性能比较。可以看出, 对于 R, G, B 各分量的 PSNR 值, 本文算法都优于文献 [6] 算法, 那么合成后的插值图像的平均 PSNR 值也高于文献 [6]。

表 1 文献 [6] 和本文算法插值灰度彩色图像性能比较

评价标准	放大 2 倍的四幅灰度插值图像						放大 4 倍的四幅灰度插值图像					
	lena		pepper		football		lena		pepper		Airplane	
	文献 [6]	本文	文献 [6]	本文	文献 [6]	本文	文献 [6]	本文	文献 [6]	本文	文献 [6]	本文
MSE	28.3187	26.2885	27.289	25.7719	26.3288	25.7078	34.0102	30.3736	33.3249	30.8177	36.7525	33.4086
PSNR	33.6101	33.9331	33.7709	34.0193	33.9265	34.0302	32.8147	33.3058	32.9031	33.2428	32.4779	32.8922
PESNR	30.2786	30.4035	28.9707	29.0345	31.1600	31.1854	30.7775	30.9787	29.6139	29.6958	29.3685	29.6178

图 5 显示 Lena 256×256 灰度图进行 5 种插值算法放大 2 倍裁剪帽檐和面部后的图像视觉效果。图 6 显示进行 5 种插值算法插值放大 2 倍文字图像“天”的视觉效果。从图 5、6 分析, Nearest 和 Bilinear 得到的插值图像在边缘有很明显的锯齿现象, Bicubic 有显著改进。本文算法视觉优于 Nearest,

Bilinear 和 Bicubic 算法, 与文献 [6] 算法视觉效果相当。

表 3 显示五种插值算法分别连续插值处理 100 幅 Lena 和 pepper 彩色图像序列从 128×128 到 256×256 的平均 CPU 处理时间 (s/图像) (实验条件是 P4 CPU 2.66GHz, 内存 256M, 编译系统 Matlab7.0)。

从实验结果分析,本文算法与文献[6]有相似的视觉质量和相近的 PSNR 值,略高的 PESNR 值,但 CPU 处理插值的平均时间较小,介于 Bilinear 和 Bicubic 之间,小于文献[6]。因此,本文算法不但有较好的视觉质量,而且有较小的算法复杂度和相对更低的计算复杂性及低平均 CPU 处理时间。

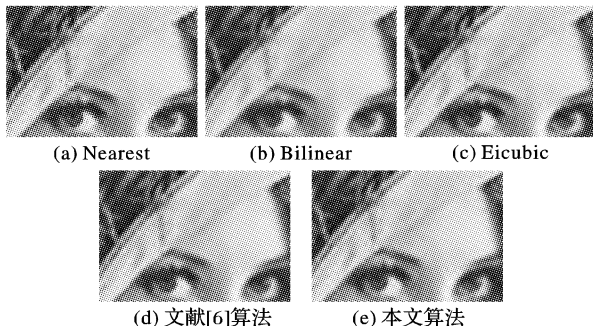


图 5 Lena 局部 5 种算法插值放大 2 倍视觉效果比较

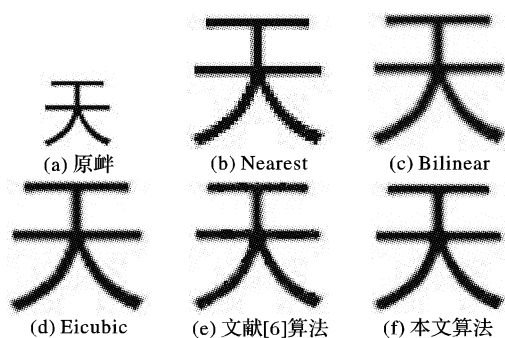


图 6 放大二倍的文字图像“天”的视觉效果

表 2 插值放大 2 倍的彩色图像 PSNR 各分量的性能比较

PSNR 标准	256 × 256 ~ 512 × 512 彩色图像					
	lena		pepper		football	
	文献[6]	本文	文献[6]	本文	文献[6]	本文
R	34.5832	34.629	34.0111	33.9488	34.3179	34.2877
G	34.2338	34.4307	34.4621	34.7271	34.1314	34.2877
B	33.3759	33.8034	34.1473	34.2266	33.7151	33.7803

(上接第 2879 页)

可见,本文方法不仅弥补了文献[5]只能得到人脸平面法线方向的不足,也具有有良好的估计精度。

4 结语

基于仿射变换和线性回归的 3D 人脸姿态估计方法是一种由粗到精的姿态估计方法。实验结果表明:利用该算法得到的 β 粗略估计值绝对平均误差约为 4.8976° ,最终估计值绝对平均误差约为 2.7453° ,且当人脸旋转角度在 40° 范围内时,估计误差都不超过 3° ,但超出 40° 时,误差有较明显增大;对于相同旋转角度下的不同人,姿态估计误差仅在大约 0.5° 的范围内波动。实验分析表明,该算法在较大的姿态变化范围内,具有良好的姿态估计精确度和鲁棒性。在该算法的基础上,下一步将通过 2D 人脸图像建立人脸 3D 模型,来估计人脸 3D 姿态,使算法适应“自遮挡”和多表情变化情况。

参考文献:

- [1] LI SZ, FU QD, GU L, *et al.* Kernel Machine Based Learning for Multi-View Face Detection and Pose Estimation[A]. Proceedings of 8th IEEE International Conference on Computer Vision[C]. Vancouver, Canada, 2001, 674 – 679.
- [2] LI SZ, ZHU L, ZHANG Z, *et al.* Statistical Learning of Multi-View Face Detection[A]. Proceeding of The 7th European Conference on

表 3 各种插值算法平均 CPU 处理时间 (s/图像)

彩色图像	Nearest	Bilinear	Bicubic	文献[6]	本文
Lena	0.010673	0.203708	0.24701	0.23585	0.226913
Pepper	0.010692	0.190914	0.252559	0.268894	0.235039

4 结语

基于边缘自适应最大相关性快速图像插值方法利用了图像边缘多个方向的最大相关性,在降低计算复杂度的同时,插值生成的图像边缘更清晰,消除了插值图像产生的锯齿,且具有自适应性和较好的视觉质量,保持了图像边缘和纹理特性。本文算法更大的优点是插值速度快,且算法复杂度小,计算量不大。

参考文献:

- [1] 王耀南,李树涛,毛建旭. 计算机图像处理与识别技术[M]. 北京: 高等教育出版社, 2001.
- [2] KEYS RG. Cubic convolution interpolation for digital image processing[J]. IEEE Transactions on Acoustics, Speech, Signal Processing, 1981, 29(6): 1153 – 1160.
- [3] BATTIATO S, GALLO G, STANCO F. A locally-adaptive zooming algorithm for digital images[J]. Elsevier Image Vision and Computing Journal, 2002, 20(11): 805 – 812.
- [4] Li X, ORCHARD MT. New edge-directed interpolation[J]. IEEE Transactions on Image Processing, 2001, 10(10): 1521 – 1527.
- [5] 陈建辉,王博亮,徐中佑,等. 一种自适应最大相关性数字图像插值算法[J]. 厦门大学学报(自然科学版), 2005, 44(3): 355 – 358.
- [6] CHEN MJ, HUANG CH, LEE WL. A fast edge-oriented algorithm for image interpolation[J]. Image and Vision 2005, 23(9): 791 – 798.
- [7] 刘晓松,杨新,汪进. 基于统计特征的彩色图像快速插值方法[J]. 电子学报, 2004, 32(1): 29 – 33.
- [8] MUKHERJEE J, PARTHASARATHI R, GOYAL S. Markov random field processing for color demosaicing[J]. Pattern Recognition Letters, 2001, (22): 339 – 351.

Computer Vision (4)[C]. 2002. 67 – 81.

- [3] CHOI KN, CARCASSONI M, HANCOCK ER. Recovering Facial Pose with the EM Algorithm[J]. Pattern Recognition, 2002, 35(10): 2073 – 2093.
- [4] YOSHINOBU EBISAWAL. Face Pose Estimation Based on 3D Detection of Pupils and Nostrils[A]. VECIMS 2005 IEEE International Conference on Virtual Environments, Human-Computer Interfaces, and Measurement Systems Giardini Naxos[C]. Italy, 2005.
- [5] YAO P, EVANS G, CALWAY A. Using Affine Correspondence to Estimate 3D Facial Pose[A]. Proceedings of the IEEE International Conference on Image Proceeding[C]. Thessaloniki, 2001, Vol 3: 919 – 922.
- [6] HU YX, CHEN LB, ZHOU Y, *et al.* Estimating Face Pose by Facial Asymmetry and Geometry[A]. Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'04)[C]. IEEE, 2004.
- [7] DEND J, LAI F. Region-based Template Deformation and Masking for Eye-feature Extraction and Description[J]. Pattern Recognition, 1997, 30(30): 403 – 419.
- [8] SHI J, TOMASI C. Good Features to Track[A]. Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition[C]. Seattle, WA, 1994. 593 – 600.