

基于边缘的实时图像缩放算法研究

刘政林 肖建平 邹雪城 郭旭

(华中科技大学电子科学与技术系, 武汉 430074)

摘要 传统的基于图像边缘的插值算法由于自身的复杂性而很少用于实时图像处理, 而不基于边缘的插值算法由于低通滤波的效应通常会使插值后的图像出现边缘模糊, 影响视觉效果。针对上述问题, 提出了一种基于边缘的自适应图像插值算法。该算法以相邻像素的二阶差分为基础, 自动选择与目标像素点相对应的源像素组进行插值运算。在 Matlab 环境中实现了各种图像插值算法, 然后对插值结果进行了主客观评价, 并对各种算法获得的图像进行了边缘检测及复杂度分析。同时, 设计了本文算法的硬件实现结构, 并用 Verilog 语言进行描述, 综合出目标代码, 最后通过 FPGA 验证。研究表明, 运用该算法插值获得了边缘清晰的目标图像, 且该算法复杂度低, 便于硬件实现。因此, 该算法能有效地实现图像插值, 适应于实时条件下的图像缩放。

关键词 实时图像处理 自适应算法 图像插值 图像边缘

中图分类号: TP391.41 文献标识码: A 文章编号: 1006-8961(2008)02-0225-05

Edge-based Algorithm of Real-time Image Resizing

LU Zheng-lin, XIAO Jian-ping, ZOU Xue-cheng, GUO Xu

(Department of Electronic Science and Technology, Huazhong University of Science and Technology, Wuhan 430074)

Abstract Due to their complexity, traditional edge-based algorithms of image interpolation are hardly used for real-time image processing. And non-edge based interpolation algorithms usually lead to the target images blurring because of their low-pass effect, so the visual effect of the interpolated images is unpleasant. In this paper, a novel adaptive edge-based algorithm of image interpolation is proposed to eliminate the defect. According to the second-order difference of the consecutive pixels, the source pixel group is automatically selected to compute the corresponding target pixel gray value. Several algorithms of image scaling are implemented in Matlab. Then the visual appearance and the edges of target images are compared, and the complexity of the algorithms is analyzed. The implementation circuit architecture for our proposed algorithm is devised and the algorithm is realized by Verilog HDL. Finally, the synthesized code is verified by downloading to FPGA on the evaluated board. The results show that our proposed algorithm achieves vivid edges of target images, and has lower computation complexity and easy for hardware implementation. So the algorithm can effectively interpolate images and adapts to real-time image resizing.

Keywords real-time image processing, adaptive algorithm, image interpolation, image edge

1 引言

图像缩放广泛应用于医学、消费电子、遥感等领域的图像处理与显示中。图像缩放的算法很多, 根据算法是否利用图像的边缘信息可将其分为两类。

第 1 类是不基于边缘的图像缩放算法, 在这类算法中, 新得到的像素值与图像的边缘信息无关, 只与其位置和其邻域的像素值有关。这类算法常用的有最近邻域法、双线性插值、双三次插值、样条插值等算法。这些经典的算法存在一些不足^[1], 通常低阶算法会出现“马赛克”现象, 而高阶算法不仅计算复

收稿日期: 2006-03-10 改回日期: 2006-10-09

第一作者简介: 刘政林 (1968~), 男, 副教授, 硕士研究生导师。2002 年获华中科技大学微电子学与固体电子学博士学位。主要研究方向为超大规模集成电路设计、数字图像处理。E-mail: liuzhenglin@21cn.com

杂,而且由于“振铃”效应会使图像边界模糊。第 2 类是基于边缘的图像插值算法,这类算法新得到的像素点的灰度值与源图像的边缘相关,可得到比第 1 类算法更满意的视觉效果,但它们算法复杂,很少应用于实时条件下的图像缩放^[2,3]。本文提出的图像插值方法是一种用于实时条件下的基于边缘的自适应图像插值算法。该算法根据相邻源像素的二阶差分值自动选取一组源像素,再根据二阶牛顿插值公式获得目标像素的值。该算法不仅获得了比不基于边缘插值算法更满意的视觉效果,而且复杂度比不基于边缘的双三次插值算法更低,远低于传统的基于边缘的插值算法。

2 基于图像边缘插值的原理

传统的基于边缘的插值算法^[3]通过检测源图像的边缘,由图像的边缘信息对图像的不同区域采用不同的插值方法。相比不基于边缘的插值算法,不但要事先对图像进行边缘检测,而且要存储图像的边缘信息,其复杂性是显而易见的。因此,这种传统的插值方法一般不用于实时图像处理。

在实时条件下对图像进行缩放处理,通常对 2 维图像依次在水平方向和垂直方向分别进行插值处理。将源图像在某一方向(如水平方向)的像素序列看作是一组等距节点,目标图像在同一方向上的任意像素点的值可由某些节点通过插值公式求得。除边界像素点外,目标图像中任意像素点与源图像像素点的位置关系如图 1 所示,任何目标像素点 N 均能找到与之对应的源像素点 f_0, f_1, f_2, f_3 , 其中 t 为目标像素点 N 与源像素点 f_0 间的距离, $1 \leq t \leq 2$ 。

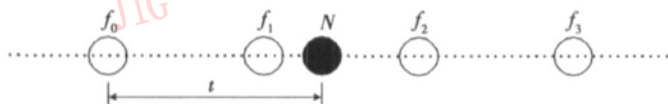


图 1 目标像素与源像素的位置关系

Fig 1 The relationship between target and source pixels

若采用不基于边缘的插值算法,如最近邻插值和双线性插值,目标像素点 N 只与源像素点 f_1, f_2 有关;若采用双三次算法,目标像素点 N 的灰度值与源像素点 f_0, f_1, f_2, f_3 有关,上述关系无论像素点 N 是否处于图像的边缘均成立。这些算法的共同点是不判断源图像中各像素点的相关性,结果不同程

度地导致了包含丰富信息的图像边缘模糊。而各像素点间的相关性隐含在图像本身,只要选择合适的参数便能得到这一重要特性。

考察源像素点 f_0, f_1, f_2, f_3 之间的相关性,将其分成两组源像素点 $I(f_0, f_1, f_2)$ 和 $II(f_1, f_2, f_3)$, 分别求出它们的二阶差分如下:

$$\Delta^2 f_0 = f_2 - 2f_1 + f_0 \quad (1)$$

$$\Delta^2 f_1 = f_3 - 2f_2 + f_1 \quad (2)$$

在此,将相邻 3 个点的二阶差分的绝对值作为衡量该 3 个点的相关性大小的标准,绝对值越小,相关性越大,反之亦然。图 1 中,目标像素点 N 的值由一组相关性较大的源像素通过插值比采用相关性较小的一组更为合理。这是因为相关性大,表明该组各像素处在同一图像区域的可能性越大。

通过比较两组源像素点的相关性,获得计算目标像素 N 的一组源像素,采用二阶插值公式即可求得目标像素 N 的灰度值。由于二阶差分已经求出,在各种经典插值公式中牛顿插值公式必须用到各阶差分,故采用二阶牛顿插值公式完成目标像素点的计算。这样可大大简化硬件电路的设计,而且不影响插值图像的视觉效果。

由此基于图像边缘的插值公式如下:

$$N_2 = \begin{cases} f_0 + \Delta f_0 t + \frac{\Delta^2 f_0}{2} t(t-1) & |\Delta^2 f_0| \leq |\Delta^2 f_1| \\ f_1 + \Delta f_1(t-1) + \frac{\Delta^2 f_1}{2} (t-1)(t-2) & |\Delta^2 f_0| > |\Delta^2 f_1| \end{cases} \quad (3)$$

$$\Delta f_i = f_{i+1} - f_i \quad i = 0, 1 \quad (4)$$

通过比较 $|\Delta^2 f_0|$ 和 $|\Delta^2 f_1|$ 的大小,选择合适的源像素点计算出目标像素的灰度值 N_2 。

3 实验结果

将本文算法及几种经典插值算法在 Matlab 环境中编程实现,对比各种插值算法处理图像的主观视觉效果、峰值信噪比及边缘信息,对各种算法进行主客观评价。

3.1 图像插值结果对比

对于灰度均一的区域,大多数算法都可获得很接近的视觉效果,而在图像边缘区域,由于灰度变化率大,不同的算法效果差别很大。为此,将着重分析图像边缘区域的缩放效果。

图 2(a)是分辨率为 512×512 的 Lena 原始图

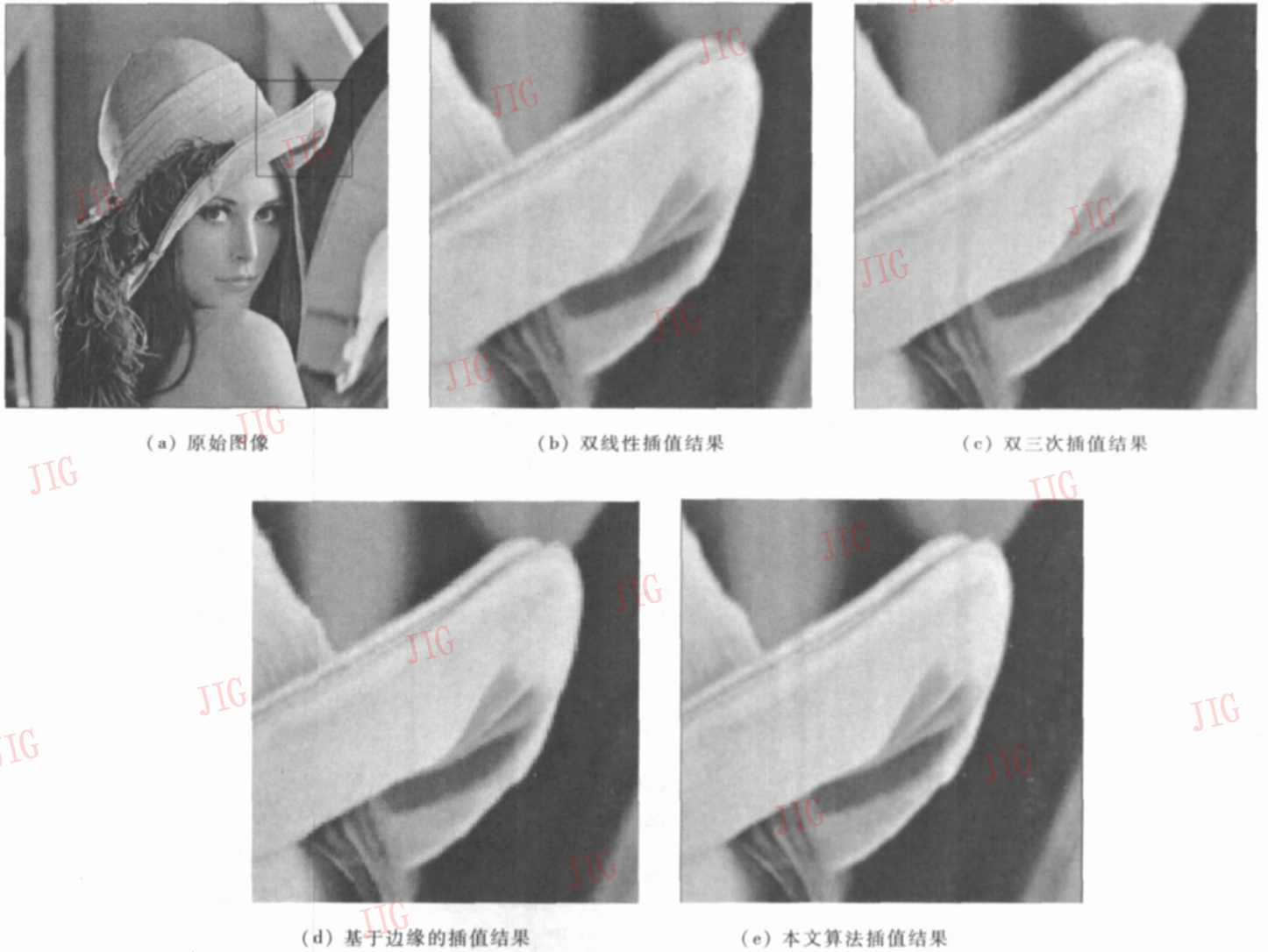


图 2 各种不同算法缩放图像效果

Fig. 2 Visual effect of resizing image by different algorithm

像, 方框标示的图像边缘区域大小为 120×120 。图 2(b) ~ (d) 分别为图 2(a) 方框区域采用不同的缩放算法得到的放大图像, 其大小均为 512×512 。由图可见, 图 2(b) 的帽沿区域出现了锯齿现象, 整体视觉效果模糊; 图 2(c) 较图 2(b) 而言, 图像整体效果平滑, 视觉效果较好, 但帽沿处出现了“振铃”现象, 轮廓不清晰, 而图 2(e) 的图像边缘轮廓清晰, 整体效果层次分明; 图 2(d) 在主观视觉效果上与图 2(e) 差异并不明显。

客观评价图像缩放算法优劣的一个重要指标是峰值信噪比 (PSNR), 其计算公式^[4]如下:

$$PSNR(F_0, F) = 10 \lg \frac{255^2}{\frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N [F(i, j) - F_0(i, j)]^2} \quad (5)$$

式中, F_0 为原始图像, F 为原始图像由某一算法放

大后, 再经相应算法缩小到与原始图像相同大小的图像。对图 2(b) ~ (d) 分别采用相对应的算法对各图像进行缩小处理, 再根据式 (5) 计算, 得到双线性插值、双三次插值及传统的基于边缘的插值处理此边缘区域的信噪比分别为 46.183 2dB、48.077 2dB、51.835 3dB, 而本文算法获得的信噪比为 49.338 6dB, 明显高于不基于边缘的双线性和双三次算法。尽管传统的基于边缘的插值获得了比本文算法更高的信噪比, 但其复杂度在实时条件下是无法接受的。

3.2 缩放图像的边缘分析

双线性插值算法有明显的低通效应, 导致高频信息丰富的图像边缘产生模糊, 而双三次插值能较好地保持图像的边缘特性, 但会出现“振铃”现象^[5]。本文提出的基于边缘的自适应图像插值算法在计算目标像素的灰度时, 根据源像素组二阶差

分最小的原则,避免了在图像边缘两侧不同灰度区域同时选取源像素参与运算,从根本上抑制了“振铃”效应的出现。

对目标图像进行边缘检测可以明显看到这一现象, Marr和 Hildreth提出的拉普拉斯边缘检测算子 $\nabla^2 G$ 被誉为最佳边缘检测器之一^[6]。对图 2(c)和图 2(d)采用 $\nabla^2 G$ 进行边缘检测,尺度空间常数均

取 2.0 所获得的边缘如图 3(a)和图 3(b)所示。图中箭头所指为 Lena 图像的帽沿。图 3(a)为双三次插值进行边缘检测的效果图,在帽沿的外侧形成了一条明显的虚边缘,而图 3(b)有效地避免的这一现象的出现(帽沿外侧有几个孤立的点,这是由于源图像的噪声所致)。

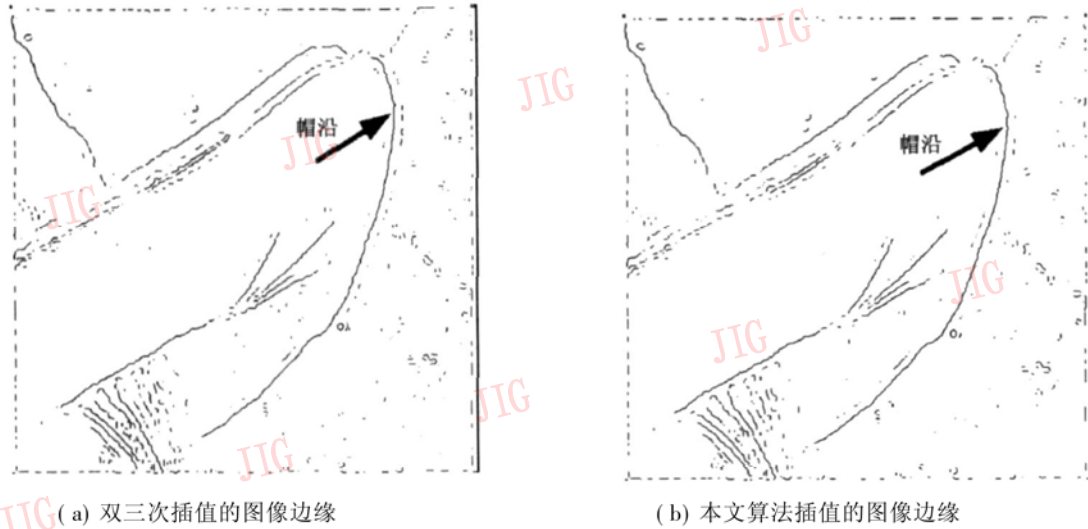


图 3 不同插值算法的图像边缘比较
Fig 3 Comparison of image edge by different algorithm

4 算法硬件实现与验证

在评价复杂算法硬件实现时,对于常数参与的加法和乘法运算占用的硬件资源可以忽略不计(例如某一未知量与 - 常数相乘,完全可以通过移位实现,而无需设计乘法器实现运算)。对于本文提出的自适应插值算法(如式(3)),尽管在 $|\Delta^2 f_0|$ 和 $|\Delta^2 f_1|$ 相对值不同的情况下表达式并不一样,但实际上在这两种情况下只是相应的输入条件不同,两种情况可统一按下式处理。

$$N_2 = f + \Delta f \cdot t + \frac{\Delta^2 f}{2!} \cdot t \cdot (t - 1) \quad (6)$$

在 t 已知的情况下,式(6)中,计算 $\Delta f \cdot t$ 需一次乘法运算, $\frac{\Delta^2 f}{2!} \cdot t \cdot (t - 1)$ 需两次乘法运算,而计算 Δf 和 $\Delta^2 f$ 分别需一次加法和两次加法运算,然后式(6)各项相加需 2 次加法运算。因此,本文方法只需 3 次乘法和 5 次加法即可完成一次插值运算。同样,双三次插值计算一目标像素点需 5 次乘法和 10 次加法运算,对比这两种算法,本文算法约减少

40% 的计算量。而最近邻域插值根据已知参数 t 直接获取像素值,双线性插值需 2 次乘法和 1 次加法运算,三次样条插值^[4]需 6 次乘法和 9 次加法运算。对于传统的基于边缘的插值算法,由于要对原图像进行边缘检测,并保存边缘信息,然后根据边缘信息在图像的不同区域选取不同的插值算法,因此,其复杂度要远远超出上述各种算法。

在实时条件下用 FPGA (field programmable gate array) 实现本文算法的数据流图如图 4 所示。系统的输入 f_0, f_1, f_2, f_3 由串联存储器 FIFO (first-in first-out) 或随机存储器提供,通过并行的加法运算即可得到一阶差分 $\Delta f_0, \Delta f_1, \Delta f_2$ 同样的方法得到二阶差分 $\Delta^2 f_0, \Delta^2 f_1$, 比较 $\Delta^2 f_0, \Delta^2 f_1$ 的大小产生的逻辑量作为 3 个多路选择器 (MUX) 提供判断条件,多路选择器自动选择一组值参与式(3)所示的运算 (\sum), 最终得到插值结果 N_2 。

用 Verilog HDL 语言对算法进行描述,在相应工具中进行综合,将得到的目标代码下载至如图 5 所示的评估板的 FPGA 中进行验证,在 LCD (liquid crystal display) 屏上得到了仿真结果完全一致的目标图像。

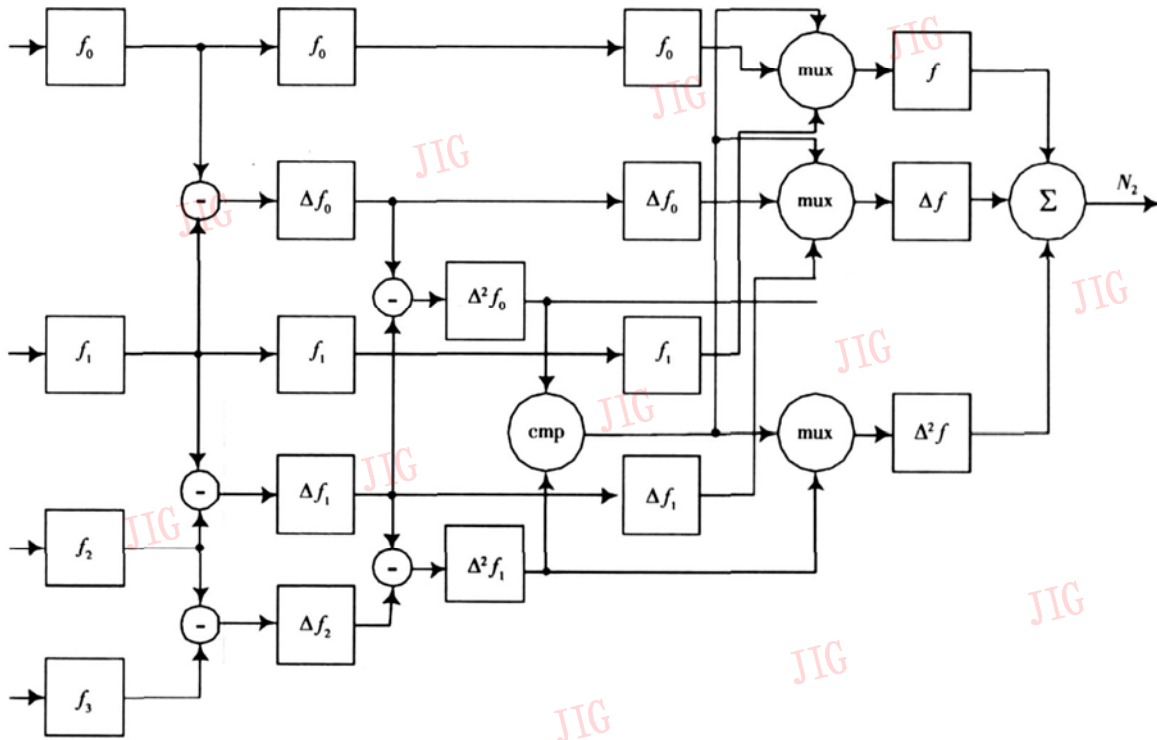


图 4 算法实现信号流程图

Fig 4 Signal flow of the algorithm's implementation



图 5 算法的 FPGA 验证

Fig 5 Algorithm's implementation by FPGA

5 结 论

提出了一种基于边缘的图像插值算法, 并采用三级流水线将该算法成功的应用于实时图像处理中。实验结果表明, 采用该算法进行图像缩放可得到比传统的不基于边缘的双三次插值算法更满意的视觉效果和更高的信噪比。而且该算法具有比双三

次插值算法更低的复杂度, 便于硬件实现。综上所述, 该算法能有效实现图像缩放, 尤其适合于实时条件下的图像处理。

参考文献 (References)

- 1 Liu Zhijun, Cai Chaq, Peng Xiaoming *et al*. A novel regularized image interpolation algorithm based on genetic algorithm [J]. Journal of Image and Graphics, 2004, 9(8): 935~ 940 [刘志军, 蔡超, 彭晓明等. 一种新颖的基于遗传算法的正则化图像插值方法 [J]. 中国图象图形学报, 2004, 9(8): 935~ 940]
- 2 Jensen K, Anastassiou D. Subpixel edge localization and the interpolation of still images [J]. IEEE Transactions on Image Processing 1995, 4(3): 285~ 295
- 3 Xin L, Orchard, M T. New edge-directed interpolation [J]. IEEE Transactions on Image Processing 2001, 10(10): 1521~ 1527.
- 4 Han J K, Baek S U. Parametric cubic convolution scaler for enlargement and reduction of image [J]. IEEE Transactions on Consumer Electronics 2000, 46(2): 247~ 256
- 5 Saif Zahir, Rabab Ward. A new edge preserving binary images resizing technique [J]. Journal of Circuits, Systems and Computers 2002, 11(3): 247~ 257.
- 6 Rafael C Gonzalez, Richard E Woods. Digital Image Processing Second Edition [M]. New Jersey, USA: Prentice Hall 2002