

# 基于动态阈值的视频对比度调节的电路设计

史再峰, 刘志贵, 姚素英

(天津大学 电子信息工程学院, 天津 300072)

摘要: 提出了基于动态阈值的视频对比度调节算法, 并通过 Verilog HDL 进行硬件描述, 采用 SMIC 0.18 $\mu\text{m}$  的 CMOS 工艺标准单元库进行逻辑综合及布局布线, 最后, 进行了后级仿真。

关键词: 图像增强 对比度调节 动态阈值

图像增强是图像处理中用于改善图像质量以及图像视觉效果的一种方法<sup>[1]</sup>。当前的一些对比度增强方法, 如全局直方图均衡化、自适应局部直方图均衡化<sup>[2][3]</sup>等方法, 能够将对对比度很差的图像调整为比较清晰的图像, 常用于医学和军事雷达或微波成像、卫星成像等领域。但是这些算法过于复杂, 不易于 ASIC 实现。在某些领域, 如视频处理或图像处理的软件或硬件设备中, 为了能够实时控制图像的对比度, 常使用一个与对比度调节系数和阈值有关的输入输出函数来修改直方图。然而当前的大多数基于修直方图修正的对比度调节算法虽可以实时地控制调节效果, 但大部分是基于经验阈值, 对较暗或较亮图像的调节效果与目的相反<sup>[4]</sup>。普通的对比度调节算法通常只采用一条曲线算法实现, 采用此曲线在增大对比度(增强)时可以获得较好的调节效果, 但在减小对比度(反增强)时却很难达到较好的调节效果<sup>[5]</sup>。本文的算法可以调节任何特殊的图像, 并且在增大或减小对比度时都能获得良好的调节效果。

## 1 基于动态阈值的对比度调节

### 1.1 对比度调节的直方图表示

设对比度调节系数为  $C$  ( $C$  为大于等于  $-1$  的实数, 大于  $0$  表示增大, 小于  $0$  表示减小)。设阈值为  $Y_{th}$ , 最大亮度为  $Y_{max}$ , 最小亮度为  $Y_{min}$ , 输入的亮度为  $X$ , 输出的亮度为  $Y$ 。

以亮度为横坐标, 各个亮度在整幅图像中出现的概率(或数量)为纵坐标, 得到图像信息的直方图。低对比度图像的像素集中在一个比较窄小的区域, 高对比度图像的像素分散于最小亮度和最大亮度之间, 这两部分的像素统计概率是相近的, 这样可以形成视觉上的亮度对比。增大对比度需要将小于阈值的亮度向最小亮度方向扩展, 而大于阈值的亮度向最大亮度方向扩展, 如图 1 中的右图。减小对比度则将亮度向阈值处收缩(如图 2 中的右图)。要获得很好的调节效果必须有一个合适的阈值和合适的亮度变化曲线。

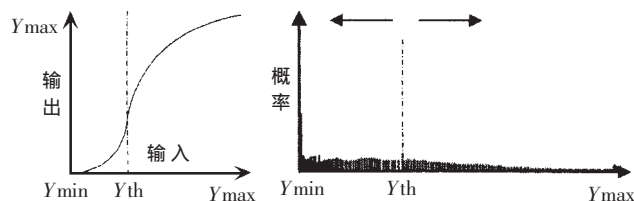


图 1 对比度增大曲线及直方图

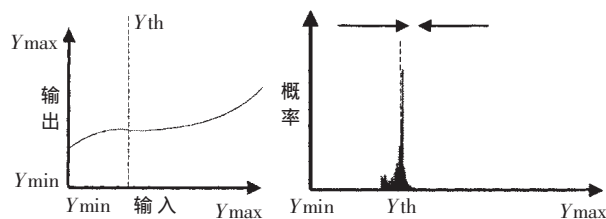


图 2 对比度减小曲线及直方图

### 1.2 基于动态阈值的对比度调节算法

调节对比度时, 分别对同一幅图像的增大对比度和减小对比度采用两种算法的输入输出曲线, 如图 1 (增大)和图 2(减小)中的曲线。

从图 1 看, 当对比度增大时, 输入亮度离阈值越远部分速率越小, 在曲线中表现为这部分斜率变得越小。从图 2 看, 当对比度减小时, 输入亮度离阈值越远部分速率越大, 在曲线中表现为这部分的斜率变得越大。这样的两条输入输出曲线既可以使调节效果明显, 又可以在一定程度上遏制亮度合并的情况。

增大对比度时, 曲线算法的合成如图 3 所示。直线为  $Y=X$  时, 曲线小于阈值部分是将  $Y_{min}$  和  $Y_{th}$  对应到曲线  $(x-x^3)$  在  $0\sim 1$  之间的这部分, 而大于阈值部分是将  $Y_{th}$  和  $Y_{max}$  对应到曲线  $(x^3-x)$  在  $0\sim 1$  之间的部分。曲线的最大幅度(弯曲程序)则由对比度调节系数决定。用直线减去曲线就可以得到增大对比度的合成曲线。

总的增大对比度的算法如公式 (1), 括号中表示条件:  $(C \geq 0 \checkmark Y_{min} < Y_{th} < Y_{max})$

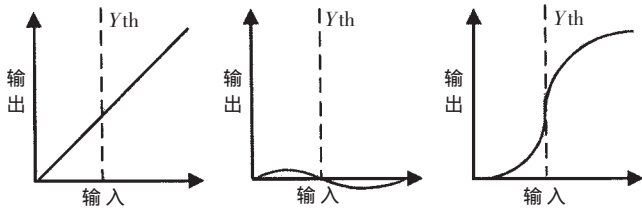


图3 增大对比度的曲线合成

$$Y = \begin{cases} X - C \times \left[ \left[ \frac{X - Y_{\min}}{Y_{\text{th}} - Y_{\min}} \right] - \left[ \frac{X - Y_{\min}}{Y_{\text{th}} - Y_{\min}} \right]^3 \right] \times (Y_{\text{th}} - Y_{\min}) & (X \leq Y_{\text{th}}) \\ X - C \times \left[ \left[ \frac{X - Y_{\text{th}}}{Y_{\max} - Y_{\text{th}}} \right] - \left[ \frac{X - Y_{\text{th}}}{Y_{\max} - Y_{\text{th}}} \right]^3 \right] \times (Y_{\text{th}} - Y_{\min}) & (X > Y_{\text{th}}) \end{cases} \quad (1)$$

减小对比度时,曲线算法的合成如图4所示,直线算法是  $Y = X + (X - Y_{\text{th}}) \times C$ , 曲线的符号与增大对比度时相反,数值大小的算法和增大对比度时一样。同样用直线减去曲线即可得到减小对比度的合成曲线。由于当对比度调节系数为0时是一条斜率为45°的直线(输出的亮度和输入的相同), 而对比度调节系数为-1时是一条斜率为0°的直线(输出的亮度全等于阈值), 所以,要采用对比度系数变换将使曲线的幅度由0增大,再减小到0,其变化关系见式(2)中的  $T_c$ 。

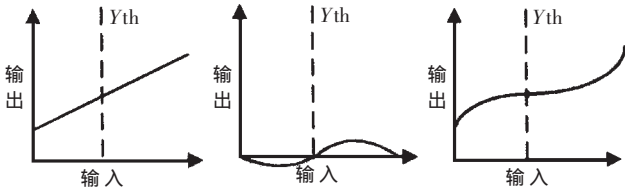


图4 对比度减小的曲线合成

总的减小对比度的算法如公式(2):

$$Y = \begin{cases} X + (X - Y_{\text{th}}) \times C - T_c \times \left[ \left[ \frac{X - Y_{\min}}{Y_{\text{th}} - Y_{\min}} \right] - \left[ \frac{X - Y_{\min}}{Y_{\text{th}} - Y_{\min}} \right]^3 \right] \times (Y_{\text{th}} - Y_{\min}) & (X \leq Y_{\text{th}}) \\ X + (X - Y_{\text{th}}) \times C - T_c \times \left[ \left[ \frac{X - Y_{\text{th}}}{Y_{\max} - Y_{\text{th}}} \right] - \left[ \frac{X - Y_{\text{th}}}{Y_{\max} - Y_{\text{th}}} \right]^3 \right] \times (Y_{\text{th}} - Y_{\min}) & (X > Y_{\text{th}}) \end{cases}$$

$$T_c = \begin{cases} C & (-0.5 \leq C \leq 0) \\ -1 \times (C + 1) & (-1 \leq C \leq 0.5) \end{cases} \quad (2)$$

另外,在图像全亮时,统计阈值等于  $Y_{\max}$ , 这时如果用上述公式(1)和(2)进行调节会出现分母为0的异常,这时应当令  $Y = Y_{\max}$ ; 而在全暗时,应当令  $Y = Y_{\min}$ 。

### 1.3 视频图像的动态阈值统计

通过统计一幅图像所有像素的亮度可以得到一个统计的阈值,准确地说是期望值。统计的阈值算法如算式(3)所示:

$$Y_{\text{th}} = (Y_{\min} \times B[Y_{\min}] + Y_{\min+1} \times B[Y_{\min+1}] + L$$

$$+ Y_{\max} \times B[Y_{\max}] / W_{\text{id}} \times H_{\text{ei}} \quad (3)$$

其中的  $B[Y_{\min}], B[Y_{\min+1}], \dots, B[Y_{\max}]$  指的是一幅图像中亮度为  $Y_{\min}, Y_{\min+1}, \dots, Y_{\max}$  的像素数目,  $W_{\text{id}}$  和  $H_{\text{ei}}$  分别指图像的宽度和高度。各个亮度的像素数目除以总的像素个数 ( $W_{\text{id}} \times H_{\text{ei}}$ ) 就是这个亮度的像素概率, 所以由算式(3)得出来的  $Y_{\text{th}}$  就是由概率统计出来的阈值。

但是此方案需要  $Y_{\max} - Y_{\min+1}$  个寄存器来存储相应亮度的像素数量, 而每个寄存器的位数必须能够存储图像的像素总数, 这需要很多的乘法器, 此办法在硬件实现上占用的芯片面积很大。如果亮度用8位表示, 则统计一幅  $1024 \times 768$  的图像要使用256个20位的寄存器。

式(3)中,用亮度为  $i$  的像素总数与亮度  $i$  相乘累加起来, 就是整幅图像的亮度总和, 所要求的阈值也就是整幅图像的平均亮度。这样只需要将整幅图像的亮度累加起来再除以整幅图像的像素数量即可得到阈值, 不需要大量的寄存器组。

从实现视频对比度调节的电路结构来分析, 如果先对每一帧图像进行阈值统计, 然后再用这个阈值对本帧图像进行对比度调节时, 需要缓冲一帧视频图像, 这就要引入大量的寄存器或RAM来缓冲图像。根据人类生理视觉模型, 要使人眼睛将一帧一帧的图像看成是连续的, 则1秒钟至少要播放24帧以上, 而且前后帧之间应当具有很强的相关性。例如MPEG-2编码采用的帧间预测编码技术就是利用了视频前后帧之间的强相关性<sup>[5]</sup>。正是这种帧间相关性使得前后帧的阈值在大部分情况下基本一致。当出现帧间跳变的情况时, 只会造成一帧图像的调整效果不太理想, 而人眼是觉察不出来的。所以在实现视频硬件电路时, 本文采用上一帧图像的阈值进行调节。

## 2 对比度调节的电路设计

### 2.1 对比度调节的系统架构

整个电路的系统架构如图5所示。其中  $I_R, I_G, I_B$  为8位的输入RGB数据总线。  $O_R, O_G, O_B$  为8位的输出数据总线。Flag是一个1位的帧同步信号, 当出现新的一幅图像时, 产生一个时钟周期的高电平。输入的亮度和对比度控制系数  $I_{\text{Bright}}, I_{\text{Contrast}}$  为8位的数据总线, 以  $8'd128$  为中心, 大于  $8'd128$  时分别表示增大

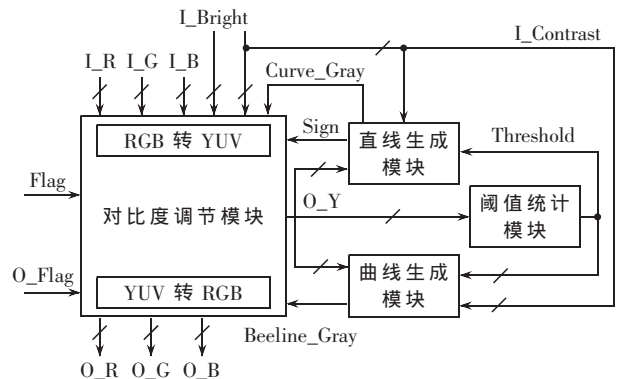


图5 系统架构

亮度和对比度，其中  $I\_Contrast$  以线性关系与上文算法中的调节系数  $C$  对应。  $O\_Flag$  为输出的帧同步信号，以用于与输出 RGB 数据同步。

直线生成模块的功能为：根据输入的亮度(图 5 中为  $O\_Y$ )产生输出亮度值  $Beeline\_Gray$ ，其中  $Beeline\_Gray$  与  $I\_Gray$  的关系在增大和减小对比度时分别对应图 3 和图 4 所示的直线。内部加入六级流水线，以便与曲线生成模块同步。

曲线生成模块的功能为：根据输入的亮度大小和输入的阈值，产生相应的亮度输出和符号。在这个模块中输入的亮度与输出的亮度呈 1.2 节所述的曲线关系。 $Sign$  为符号位用于指示对比度调节模块，当其为 1 时用  $Beeline\_Gray$  减去  $Curve\_Gray$ ，反之则相加。此模块中需要一个并行的流水线除法器，本文采用 Synopsys 的 DesignWare Block IP 中的并行流水线除法器  $DW\_div\_pipe$ 。除法器是整个电路的速度和面积的瓶颈，流水线除法器采用三级流水，曲线生成模块内部总共加入了六级流水线。

阈值统计模块的功能为：对每次输入的亮度累加，并在每幅图像的帧消隐期间计算出阈值，并输出  $Threshold$ 。在阈值统计模块内部由于需要将累加的亮度除以图像的总像素数目，所以使用了一个串行除法器。本文采用 DesignWare Block IP 中的串行除法器  $DW\_div\_seq$ 。如果图像宽度和高度是固定的，则这个模块中的串行除法器可以采用其他方式(如移位)来实现。

对比度调节模块的功能为：对外部输入的 RGB 信号 ( $I\_R, I\_G, I\_B$ )产生 YUV 信号，将 Y 信号( $O\_Y$ )送往直线生成模块、曲线生成模块和阈值统计模块，并且合成由直线生成模块和曲线生成模块输入的值，并将 YUV 还原为 RGB 信号输出，在这个模块内部由于 RGB 与 YUV 相互转换加入了两级流水线。

## 2.2 对比度调节电路实现结果及验证

用 Verilog HDL 对电路进行 RTL 描述，采用 SMIC0.18 $\mu\text{m}$  CMOS 工艺标准单元库进行逻辑综合。结果表明，整个电路(包括亮度调节模块和 YUV2RGB 模块等外围电路模块)速度可以达到 100MHz，并且通过了布局布线之后的后级仿真验证。采用 Design Compiler 2004.6 的综合时序报告如表 1 所示，综合面积报告如表 2 所示。图 6 是布局布线之后的后级仿真的波形图。

为了测试电路对视频的调节效果，本文将 Lena 图片制作成视频，然后转换成适合于用 Verilog 读取的数据，通过 Verilog 前级和后级仿真的测试，截图效果如图 7 所示。图 8 为另一段视频的调节效果截图。图 7 和图 8 中的最上面的一组图片为原图及其直方图。中间的一组

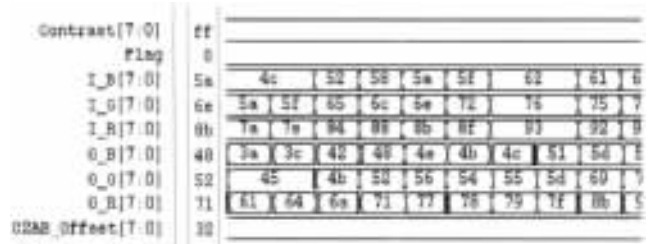


图 6 后级仿真波形图

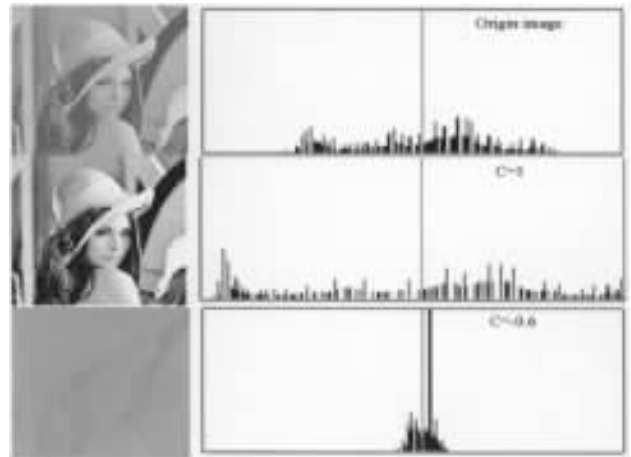


图 7 视频处理效果图

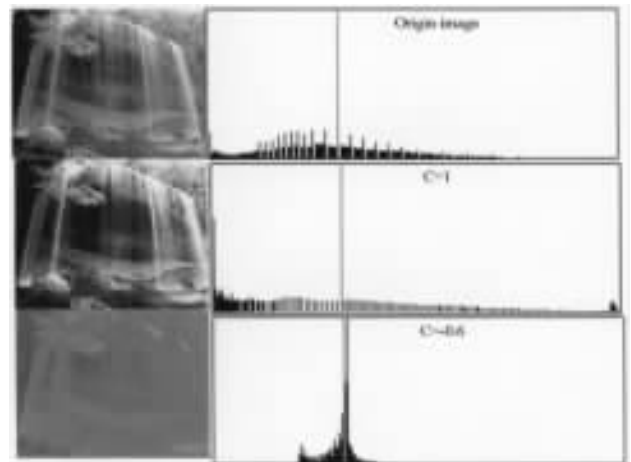


图 8 视频处理效果图

图片为对比度增大的图片及其效果， $I\_Contrast$  为 8'd255(对应于算法中的  $C=1$ )。最底下的一组为对比度减小的图片及其效果， $I\_Contrast$  为 8'd44(对应于算法中的  $C=-0.6$ )。

本算法及电路充分考虑了硬件的易实现性和人的视觉特性，针对增大和减小对比度分别采用了基于由图像统计出来的动态阈值的不同的调节曲线算法，既可以使图像的直方图得到扩展(或收缩)，又在一定程度上控制了亮度合并的情况。在最大程度上既增大(或减小)了对比度，又对图像的细节损失做出了一定的抑制。

### 参考文献

[1] 俞诗鲲, 郑建生, 曾欣. 用 CPLD 实现嵌入式平台上的

(下转第 68 页)

表 1 综合后的时序报告

关键项	路径(ns)
时钟周期	10.00
需求时间	9.21
松弛度(MET)	0.01

表 2 综合后的面积报告

属性	面积( $\mu\text{m}^2$ )
组合逻辑	194 181.937 500
非组合逻辑	57 376.570 312
总计	260 543.125 000



限值可以准确地判断出待测图像是否存在重影。

(2)与FSWM算子法类似,利用SMD算子计算Sratio,得到图4所示的折线图。

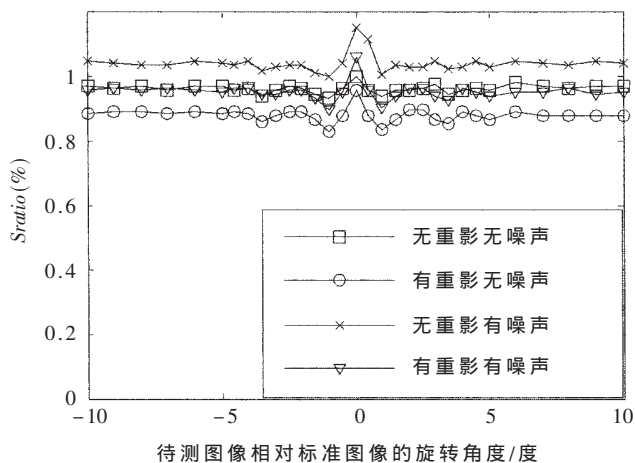


图4 SMD算法性能图

从图4中可见,当待测图像无重影时,由于噪声的影响,导致Sratio的值波动很大(表现为方框线与叉号线相差很大);当待测图像有重影时,由于噪声的影响,同样会导致Sratio的值波动较大(表现为圈号线与三角线偏差较大),因此,SMD算子法对噪声很敏感。

(3)采用互相关匹配法可得到如图5所示的结果。Corr值即待测图像与标准图像之间的互相关系数。从图5中也可看到,互相关匹配法极易受到噪声影响,因此各个折线相差较大。从图5中可以看到,在 $[-1^\circ, 1^\circ]$

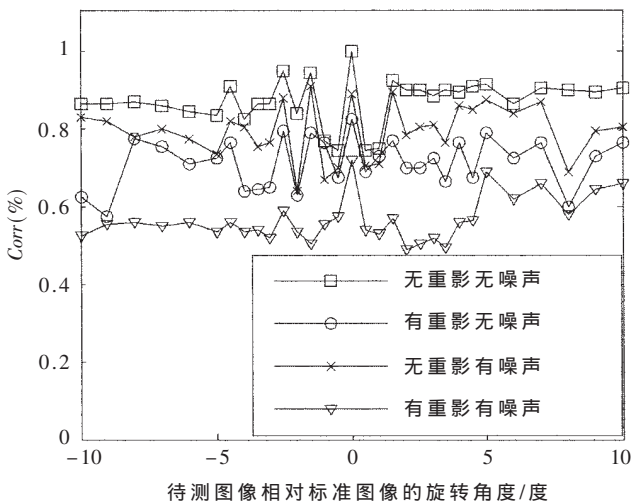


图5 互相关匹配法性能图

的小旋转角度范围内,Corr的值波动较大,反映了当待测图像相对于标准图像旋转角度较小时,配准精度有所波动。反观图3可见,FSWM算子法在同样的小角度范围内,Fratio的变化要平稳得多,可以看出,互相关匹配法受匹配精度影响较大,而本文所采用的FSWM算子法受匹配精度影响较小。

在Windows平台下,利用Matlab7.0.1实现了相应检测程序。本文采用的实验数据为300幅待测图像,图像大小为 $650 \times 300$ 像素。通过实验得到四种重影检测算法的检测性能如表1所示。

表1 四种重影检测方法检测性能表

	检测率	漏检率	虚警率	所耗时间(s)
FSWM算子法	0.945	0.03	0.025	0.23
SMD算子法	0.717	0.08	0.200	0.19
互相关匹配法	0.624	0.09	0.280	0.34
差值图像法	0.583	0.12	0.300	0.12

本文提出了基于FSWM算子的重影检测方法,并将该方法与其他两种重影检测方法进行对比,实验结果表明本文所采用的FSWM算子法检测重影准确率高,与其他传统的检测方法相比,在脉冲噪声和配准精度的鲁棒性方面有较大的改进,能较好地满足印刷品的重影检测。进一步的工作可以归纳为如下两点:

(1)改进图像预处理中所采用的图像配准方法,使得配准精度更高,速度更快。

(2)可以考虑,在FSWM算子中只用一个方向上的FSWM算子,即水平方向或垂直方向,以减少计算量。

### 参考文献

- [1] RADKE R J, ANDRA S, Al-Kofahi O et al. Image change detection algorithms: a systematic survey Image Processing[J], IEEE Transactions, March 2005: 294-307.
- [2] 曲中周. 基于图像处理的香烟小包装在线实时检测系统[D]. 南京航空航天大学硕士学位论文, 2004.5.
- [3] 刘树凤. 数码相机自动聚焦算法的研究与实现[D]. 华南理工大学硕士学位论文, 2003.2.
- [4] CHOI K S, LEE J S, KO S J. New autofocusing technique using the frequency selective weighted median filter for video cameras[J]. Consumer Electronics, IEEE Transactions on. June 1999: 820-827.
- [5] 张登银, 汤志云. 基于边缘配准的印鉴自动鉴别方法[J]. 南京邮电学院学报, 2003, (9): 25-30.

(收稿日期: 2006-09-19)

(上接第62页)

实时图像增强[J]. 电子技术应用, 2003, (12).

- [2] JEN T C, HSIEH B, WANG Sheng Jyh. Image Contrast enhancement based on intensity-pair distribution[J]. Volume 1, 11-14 Sept. 2005 Page(s): 1-913-16 Digital Object Identifier 10.1109/ICIP.2005.1529900.
- [3] Zeyuti YLI, Clznridmjit Bajij. A fast and adaptive method

for image contrast enhancement[J]. International conference on image processing(ICIP), 2004: 1001-1004.

- [4] 田捷, 沙飞, 张新生. 实用图像分析与处理技术[M]. 北京: 电子工业出版社, 1994: 71-74.
- [5] 井上诚喜, 八木伸行, 林正树等. 实用数字图像处理[M]. 北京: 科学出版社, 64-65.

(收稿日期: 2006-09-20)