

一种摄像机采集图像的二值化算法

王纪川^{1,2}, 刘宇³, 齐勇¹, 侯迪¹

WANG Ji-chuan^{1,2}, LIU Yu³, QI Yong¹, HOU Di¹

1.西安交通大学 计算机科学技术系 计算机软件研究所, 西安 710049

2.西安工程大学 计算机科学学院, 西安 710048

3.清华大学 计算机科学与技术系, 北京 100084

1.Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an 710049, China

2.School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China

3.Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

E-mail: jcwang@xpu.edu.cn

WANG Ji-chuan, LIU Yu, QI Yong, et al. Binarization algorithm of video camera collecting image. *Computer Engineering and Applications*, 2007, 43(30): 202-205.

Abstract: To record the correlative parameters of each tube in nondestructive testing automatic pipeline, the printing number and reading number at the end of the tube is needed. A key image preprocessing binarization algorithm in tube number reader system is introduced. The algorithm is called after Stamp, which is based on Stamp filter in Photoshop and video camera collecting characteristic image. It is still applied to image preprocessing in automatic vehicle license plate recognition et al.

Key words: algorithm; binary image; binarization; accumulated histogram

摘要: 在无损伤自动检测线上, 为了记录每根管材的有关参数, 需在管端进行打号、读号处理。详细介绍了管号识别系统中关键的图像预处理二值化算法。该算法称之为 Stamp 算法, 是在分析 Photoshop 图像处理软件 Stamp 滤镜功能的基础上, 结合自动检测线上采集图像的特点而得出的。Stamp 算法还可以用于车辆牌照自动识别的二值化处理等领域。

关键词: 算法; 二值图像; 二值化; 累积直方图

文章编号: 1002-8331(2007)30-0202-04 **文献标识码:** A **中图分类号:** TP391.41

在无损伤自动检测线上, 为了保证产品的质量, 要对生产的无缝钢管进行诸如超声、涡流探伤等工艺处理。由于检测线上的管材本身无任何标识, 因此所检测的管材数据无法和每根管材自动对应。为了记录每根管材的有关参数, 需在管端进行打号、读号处理。某钢管公司精整车间自动检测流水线上的工艺流程如下: …→加塞→沙带抛光→砂轮抛光→清洗→烘干→打号→读号→超声检测→分检→读号→涡流探伤→分检→读号→锯切→端正→…→成品包装。在该检测线上共应用了三套“管号识别系统”^[1], 分别与超声检测装置、涡流探伤装置、锯切装置配套使用。其中超声检测装置、涡流探伤装置检测管材的“管长”、“外径”、“内径”、“壁厚”、“缺陷位置”等关键参数^[2]; 锯切装置根据检测到的管材上的缺陷位置对管材进行锯切, 确保最后留下的管材为合格产品。

“管号识别系统”主要由图像采集、图像二值化、图像细化、图像分割和图像识别 5 部分组成。其中通过摄像机采集的管号彩色图像的二值化处理是一关键技术。经过各种途径的试验, 发现 Photoshop 图像处理软件中 Stamp 滤镜功能二值化此种图像比较有效。借助 SoftICE 工具对 Stamp 滤镜程序进行了解剖, 基本分析清楚了其工作原理, 并在此基础上针对工业现场所采

集图像的特点对该算法进行了改进, 解决了此种图像的二值化问题, 为后续的图像细化、图像分割和图像识别打下了良好的基础。

1 Stamp 算法分析

Stamp 算法处理的输入和输出图像均为彩色图像, 每个像素由 R 、 G 、 B 三字节组成, 而中间处理的图像为灰度图像, 由 g 单字节构成。Stamp 算法有两个参数, 一个参数为明暗平衡度 LDB , 其值域为 $0\sim 50$; 另一个参数为光滑度 S , 其值域为 $1\sim 50$ 。 LDB 和 S 的取值依所处理的图像的特点而定。明暗平衡对比度数组 L_Dtable 的取值参看附录(1)。根据 LDB 的取值, 用其作为下标查找 L_Dtable 数组对应的数组元素值 n , 然后将 $Table2$ 数组的前 n 个数组元素值置为 0, 其余的数组元素值置为 255, 供后续查表之用。

下述框图和图示说明中的主要变量约定为: 源图像宽度 SW , 源图像高度 SH , 扩展后图像宽度 EW 。Stamp 算法主要由 Stamp 主函数、SF2480、SF27B0、SF2930、SF2370、SF2330、SF21B0、SF2580、SF2780 等 8 个子函数和 5 个 Table 数组构成。为了论述的简便性, 所有函数的参数均不做详细介绍。

作者简介: 王纪川(1962-), 男, 博士生, 副教授, 主要研究领域为分布式系统理论及应用、图像处理和模式识别; 刘宇(1977-), 男, 博士后, 主要研究领域为计算智能、数据挖掘、图像处理; 齐勇(1957-), 博士, 教授, 博士生导师; 侯迪(1960-), 男, 副教授。

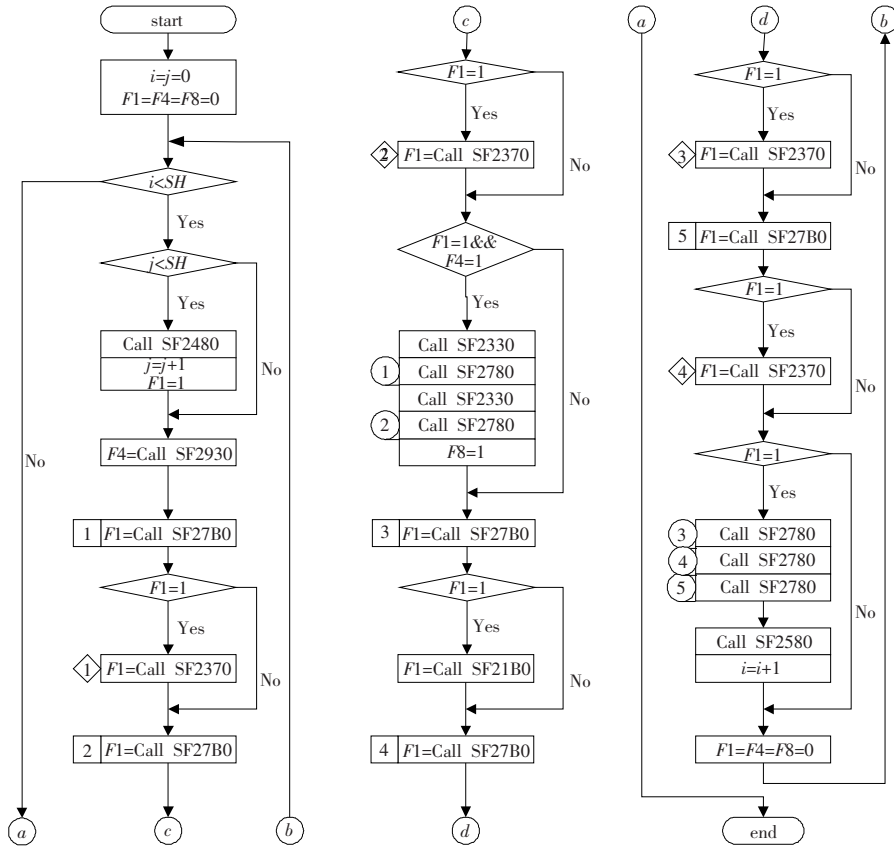


图1 Stamp 算法框图

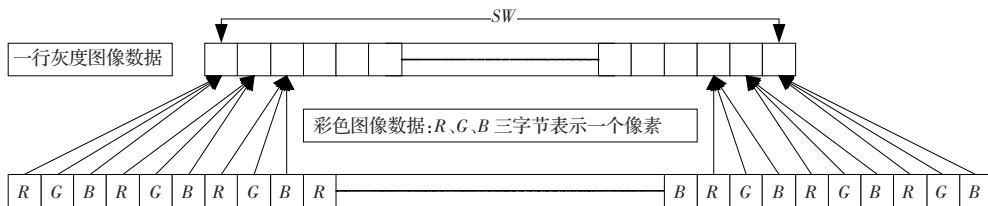


图2 一行彩色图像数据转换为灰度图像数据的处理方法

1.1 Stamp 算法主函数框图

在处理每行图像数据时, 共调用子函数 SF27B0 5 次、SF2370 4 次、SF2780 5 次, 如图 1 中方框左边的序号所示。子函数 SF27B0 根据输入的参数判断是否进行图像的行、列扩展; 子函数 SF2370 根据输入的参数判断是否进行区域像素的平均灰度值计算; 而子函数 SF2780 为一查表子函数, 5 次调用依次分别为查表 Table1、Table2、Table3、Table4 和 Table5, 至于各表的数值请参看附录。值得一提的是子函数 SF27B0, 每次调用时均有各自独立的计数器和缓冲区等参数。

1.2 Stamp 算法子函数

下面分别介绍子函数 SF2480、SF27B0、SF2930、SF2370、SF2330、SF21B0 和 SF2580。由于 SF2780 子函数是一查表函数, 本文不予介绍。

1.2.1 彩色图像数据转换为灰度图像数据

子函数 SF2480 的功能: 逐行将每像素的彩色图像数据转换为灰度图像数据。转换关系为: $g = (R + G + B + 2) / 3$, 其中 R 、 G 、 B 表示该像素的彩色数据, g 表示该像素的灰度值。该函数的功能如图 2 所示。

1.2.2 灰度图像的行和列扩展方法

子函数 SF27B0 的功能: 根据输入的参数判断是否进行图像的行、列扩展。3 种扩展方式分述如下:

(1) 每行灰度图像数据的列扩展

扩展的总列数为 $(P3) / 2 * 2$, 即左、右两端各扩展 $(P3) / 2$ 个像素。扩展后每行图像数据的长度 $EW = SW + (P3) / 2 * 2$ 。其扩展方式如图 3 所示。

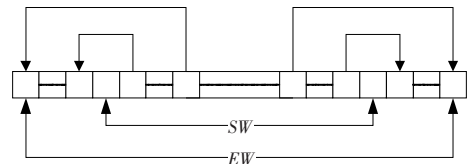


图3 一行灰度图像数据的列扩展方式

(2) 图像顶部的行扩展

当处理的图像当前行数为 $(P3) / 2$ 时进行图像顶部的扩展。一次扩展 $(P3) / 2$ 行数据。其扩展方式如图 4 所示。

(3) 灰度图像底部的行扩展

当处理的图像当前行数 $\geq SH - (P3) / 2$ 时进行图像底部的

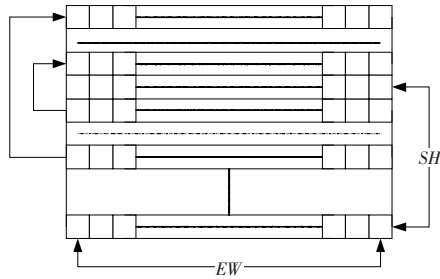


图4 灰度图像顶部的行扩展方式

扩展。一次扩展一行数据。其扩展方式如图5所示。

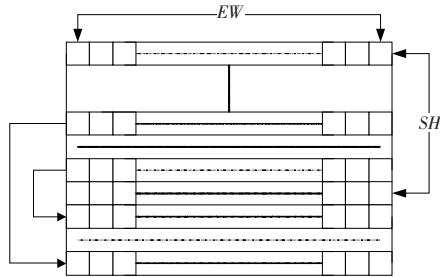


图5 灰度图像底部的行扩展方式

1.2.3 环形缓冲区

子函数 SF2930 的功能:根据输入参数判断是否有原始图像数据需要保存到环形缓冲区,并判断是否输出一行原始图像灰度数据供后续的子函数使用,如图6所示。

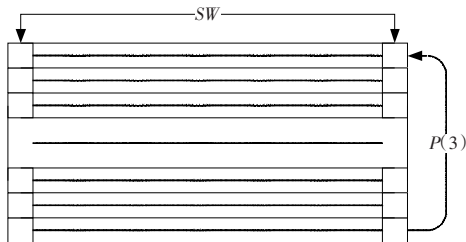


图6 环形缓冲区

1.2.4 计算灰度图像区域像素平均灰度值

子函数 SF2370 的功能:在行数为(P2)的缓存区内,求(P2)*(P2)的区域内像素的平均灰度值,然后将结果存入参数 P3 所指定的存储区。若图像区域大小为 M×M(M 为奇数),则像素平均灰度值为:

$$a(i,j) = \frac{\sum_{k=-\frac{M}{2}}^{\frac{M}{2}} \sum_{l=-\frac{M}{2}}^{\frac{M}{2}} f(i+k, j+l)}{M \times M}$$

该函数的处理方式如图7所示。

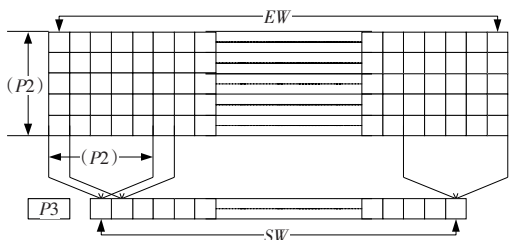


图7 像素平均灰度值的计算方式

1.2.5 计算灰度图像像素差值

子函数 SF2330 的功能:用参数 P1 所指定的存储区内的值依次减去参数 P2 所指定的存储区的值,若差大于0 则将差存入参数 P2 所指定的存储区,否则将0 存入参数 P2 所指定的存储区。该函数的处理方式如图8所示。

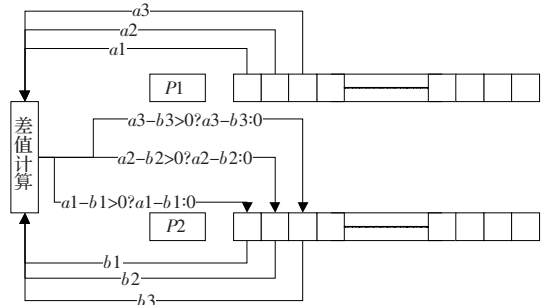


图8 计算灰度图像像素差值的方式

1.2.6 求灰度图像区域像素的累积直方图≤(P2)-1 的最大灰度值

子函数 SF21B0 的功能:在行数为(P2)的缓存区内,统计(P2)*(P2)区域内像素的直方图,然后求灰度图像区域像素的累积直方图≤(P2)-1 的最大灰度值,并将该灰度值存入参数 P3 所指定的存储区。

(1)统计区域像素的直方图

计算 L 个灰度级的区域像素的直方图 $h(l)=n_l, l=0, 1, \dots, L-1$ 。其中 n_l 为该区域中具有灰度值 l 的像素个数。直方图提供了该区域像素的灰度值分布情况^[3],如图9所示。

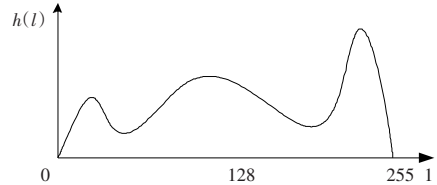


图9 区域像素的灰度值分布

(2)求累积直方图的最大灰度值

对 L 级灰度的区域像素,计算其累积直方图 $h'(l) = \sum_{i=0}^l h(i)$, $l=0, 1, \dots, L-1, h(l)$ 同上。如图10所示,即求满足下式的 l 。

$$\begin{aligned} & \operatorname{argmax}_l h'(l) \\ & \text{s.t. } h'(l) \leq (P2)-1 \end{aligned}$$

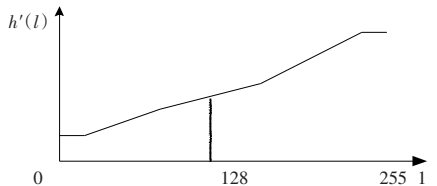


图10 累积直方图

(3)该函数的处理方式如图11所示。

1.2.7 灰度图像数据转换为彩色图像数据

子函数 SF2580 的功能:逐行将每像素的灰度图像数据转换为彩色图像数据。转换关系为:R=G=B=g。该函数的功能如图12所示。

2 实验结果

图13和图15分别为在实验室环境和自动检测线上采集

