

一种高速多功能图形处理系统的实现与应用

张毅 邢炜 周兆英

(清华大学精密仪器与机械学系,北京 100084)

摘要 介绍了一个高速多功能的二值图形处理系统,该系统主要由硬件逻辑电路组成,高速灵活地实现轮廓跟踪,细化等多种常用的二值图像处理算法,处理速度主要受硬件延时的限制,该系统可在许多领域中广泛应用,文中给出了其中的一个应用实例。

关键词: 二值图形处理系统;逻辑电路;轮廓跟踪

1 概 述

利用计算机处理二维图形时,常将图形转化为二值图像,因为在数字图像中,二值图像占有非常重要的地位。特别是在实用的图像处理系统中,以二值图像处理为基础构成的系统是很多的。目前已有多种方法用于分析,提取和识别图形中物体的结构特性,解决二值图像处理中遇到的各类问题,其中包括收缩、膨胀、细化、轮廓跟踪等。这些处理方法在许多应用中都会遇到,比如收缩、膨胀的反复组合使用就可以把二值图像中的细小成分或小孔检验出来或消除掉,因而这种处理可用于检查图形中的细小缺陷。如用于以掩膜图检查为代表的各种工业检查系统中及粒子图像分析系统中;细化就是使给定的图形线宽变细,从而提取线宽为一个像素的中心线的操作。这种处理可广泛用于指纹、电路图、染色体图的识别以及某些OCR系统中;轮廓跟踪则是提取图形边界点的过程,主要用于形状分析中。为了实现上述这些处理方法,研究者们针对不同的处理过程发表了一系列相应的算法。然而,多数算法都比较复杂,程序运行费时间,随着图形规模的增大,这一问题更为突出。一般情况下,算法程序运行所用时间主要同以下几个因素有关:

1. 图形幅面大小;
2. 图形的复杂程度;
3. 图形的线宽及扫描分辨率;
4. 计算机的运行速度;
5. 程序的优化水平。

针对这种情况,有必要设计一种高速处理系统,以提高图形处理效率。实际上这几种处理方法有许多共同的性质,它们都包含平面区域的遍历问题。根据这种相似性,有可能设计出一种实现上述各种处理方法的图形处理系统。基于这种分析,本文介绍了一种以硬件为基础的高速多功能图形处理系统。文中第二节首先介绍了有关的算法,接着在第三节中说明了各种算法

的实现与系统的设计,第四节介绍了该系统的应用实例。

2 几种图像处理算法

由扫描得到的二值图像可以被看作是分布于二维平面上分离象素的集合 BI ,任一象素都对应一数值 $P \in (0,1)$,在本文的讨论中,令象素为白色时其值为 1,象素为黑色时,其值为 0。图中任一象素 P 都被 8 个相邻象素 $P_0 \sim P_7$ 所环绕,如图 1(a)所示,形成一个 3×3 的“窗”,象素 P 位于窗的中心。如果把 8 个相邻象素编以序号图 1(b),则任一象素 P 均有一“邻域码” NC 与之对应。

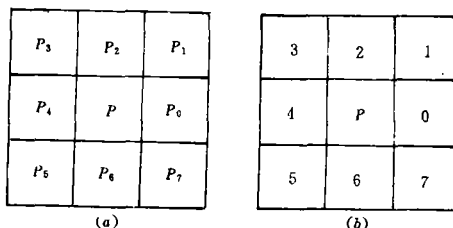


图 1 任一象素 P 及其 8 邻域

在十进制中有

$$NC = \sum_{i=0}^7 2^i P_i \quad (1)$$

其中 i 为图 1(b) 中所示 8 个相邻象素的序号, $P_i \in (0,1)$, 很明显有 $0 \leq NC \leq 255$ 。于是只要给出 P 及其对应的 NC 值即可确定该象素的特性(黑或白)及图形的状态。例如: $P=0, NC=0$, 则 P 这图形中间的一个象素,其周围皆为“黑点”;而如果 $P=0, NC=255$, 则 P 为孤立点,其周围皆为“白象素”。

2.1 轮廓跟踪、收缩、膨胀

在二值图像象素集合 BI 中可按象素的颜色分为两个子集 $W\{P=1\}, B\{P=0\}$, 即白象素子集和黑象素子集,或称为白色区域和黑色区域,则 $BI=W+B$ 。对任一象素有

- (1) 如果 $P=0$ 且 $NC=0$, 则该象素为黑色区域内部点;
- (2) 如果 $P=0$ 且 $NC \neq 0$, 则该象素为黑色区域边界点;
- (3) 如果 $P=1$ 且 $NC=255$, 则该象素为白色区域内部点;
- (4) 如果 $P=1$ 且 $NC \neq 255$, 则该象素为白色区域边界点。

由上述分析可以直接得出相应的算法。

(1) 轮廓跟踪: 如果某一象素满足 $P=0$ 且 $NC \neq 0$, 则予以保留, 否则删除(即把该象素颜色由黑变白)

(2) 收缩: 如果某一象素满足条件 $P=0$ 且 $NC=0$, 则予以保留, 否则删除。

(3) 膨胀: 如果某一象素满足条件 $P=1$ 且 $NC=225$, 则予以保留, 否则删除(即把原象素由白色变为黑色)

收缩与膨胀只是相对而言, 黑色区域的收缩即是白色区域的膨胀, 反之亦然。

2.2 细化

细化的目的是要提取能保存图形本质结构的线型图型, 因此要求细化算法应具备以下性质:

- (1) 细化结果应与原图具有拓扑等效性;
- (2) 细化结果应保持原图中图型的连续性;
- (3) 细化结果为(或接近)原图线型的中心;
- (4) 细化后线宽为一个象素;

(5) 细化结果对原图边界噪声不敏感。

围绕这几个特性,已发表了许多细化算法⁽²⁾,本节只选取一种⁽³⁾作简单说明。由于符号定义与原文有所不同,故表达形式亦有变化。这是一个并行算法,叠代运算,直至细化完成。每次叠代又分为两个步骤。首先定义变量

$A(P)$:在某一象素 P 的 8 邻域象素中,按 $P_0, P_1, P_2, \dots, P_7, P_0$ 顺序发生 $0 \rightarrow 1$ 变化的次数;

$B(P)$:在某一象素 P 的 8 邻域象素中“0”(即“黑”象素)的数目。

第一步,如果 $P=0$ 且 P 的 8 邻域象素满足如下条件则被删除;

$$(1) 2 \leq B(P) \leq 7; \tag{2}$$

$$\left. \begin{aligned} & A(P)=1 \text{ 时, } P_0+P_2+P_6=1 \text{ 且 } P_0+P_4+P_6=1; \\ (2) & A(P)=2 \text{ 时, } P_0+P_2=0 \text{ 且 } P_4 * P_5 * P_6=1; \\ & \text{或 } P_0+P_6=0 \text{ 且 } P_2 * P_3 * P_4=1. \end{aligned} \right\} \tag{3}$$

第二步,如果 $P=1$ 且 P 的 8 邻域象素满足如下条件则被删除:

$$(1) 2 \leq B(P) \leq 7; \tag{4}$$

$$\left. \begin{aligned} & A(P)=1 \text{ 时, } P_0+P_2+P_4=1 \text{ 且 } P_0+P_4+P_6=1; \\ (2) & A(P)=2 \text{ 时, } P_2+P_4=0 \text{ 且 } P_0 * P_6 * P_7=1; \\ & \text{或 } P_4+P_6=0 \text{ 且 } P_0 * P_1 * P_2=1. \end{aligned} \right\} \tag{5}$$

3 算法的实现

由第二节的内容可知,这几种算法都是基于对某一象素的 8 邻域象素的分析,故每处理一个象素,首先都要提取出它的 8 邻域信息,然后再根据算法做出逻辑判断,以确定该象素是否应被删除,如果算法比较复杂,甚至还要作叠代运算(如上述细化算法),则逻辑运算的量要成倍增加。考虑以 300 点/英寸的分辨率扫描一张 A4 幅面的图形,所生成的文件容量超过 1M 字节,即所要处理的象素的数目将超过 800 万个,可以想象这将是一项很费时的工作。对于更大规模(如 A0 幅面)的图形操作时间将更长,而对于某些具有实时性要求的场合,算法甚至可能无法应用。通过分析可以发现,在这几个算法中,重复性操作是很多的,如叠代运算,对每一点的相同操作等,这种特性是很适合于用硬件电路实现算法的。实际上,每一算法的最终结果就是确定某一象素是否应被删除(即使象素由黑变白或由白变黑),而该象素删除与否又是由其 8 个相邻象素所处的状态决定的。如前所述, $0 \leq NC \leq 255$, 故某象素的 8 邻域象素在某一时刻所具有的状态,共有 256 种可能,如果以这 256 种可能为一个集合, $S = \{0 \leq NC \leq 255\}$, 那么 S 可分解为两个子集 $S = S_1 + S_2$ 。当 $NC \in S_1$ 时,所对应的象素应被删除,而当 $NC \in S_2$ 时,对应的象素应予保留。可以认为上述几种算法就是要确定某一象素的 NC 的所属集合(S_1 或 S_2)。所以可以将上述几个算法制成不同的表格,以 NC 值作为位置标志,而将对应该 NC 值的象素删除与否的信息(0 或 1)作为存于该位置内容。这样就上述几个算法的实现变成了查表过程。这个过程可以用硬件电路实现。如图 2 所示,由 9 个 D 触发器构成一个 3×3 的阵,9 个输出端分别对应于图 1 中的 P_0, P_1, \dots, P_7, P 。由此可得到某一象素 P 有 8 邻域值 NC (以二进制表示)。将算法表存入存储器中(ROM 或 RAM),图 2 中的输出端 P_0, \dots, P_7 按顺序接至存储器地址的低 8 位,而在相应地址对应的单元中存入删除信息。随着数据时钟的到来,存储器的输出端即得到对象素的修改信息,将该信息与原象素值作逻辑运算,即可达到修改的目的。

图 3 为系统的原理图,其中 A, B 为两个存储器组,用于存放图形文件, T 为用于存放算法

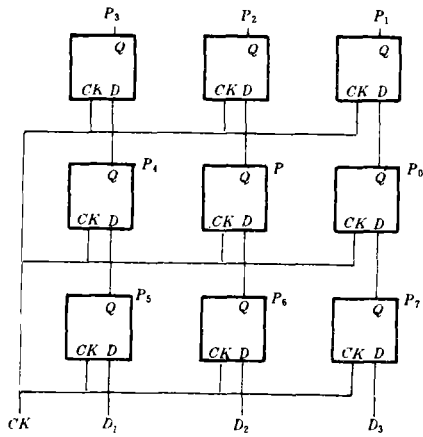


图2 3×3窗

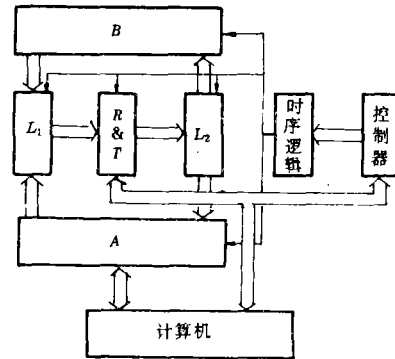


图3 系统原理图

表的存储器, R 是由若干个 D 触发器组成的阵列, 用于提取 8 邻域信息, L_1, L_2 是两个逻辑控制单元。系统工作前, 首先由计算机将图形文件按固定格式存入存储器 A, 开始运行后, 数据由 A 输出, 经 R&T 查由控制逻辑指定的算法表, 得删除信息, 将其与原始信息一同输入 L_2 , 经逻辑运算将修改结果存入存储器 B, 如需叠代运算, 只需按相似过程, 使数据再由 B 输出, 修改后存入 A, 如此过程反复进行, 直至满意为止。收缩、膨胀和轮廓跟踪三个算法占用 256 个字节, 细化算法需占用 512 个字节, 这样, 用于存储算法表的存储器只需 512 字节即可。该系统每个时钟周期可处理 8 个像素, 时钟频率只受硬件时延限制。

4 应用

上述系统可用于对实现二值图像的收缩、膨胀、轮廓跟踪和细化处理。凡是需要作这几种处理的场合, 均可利用该系统。如概述中提到的工业检查系统, 粒子图像分析, 指纹及染色体识别等。本节将以该系统在图形识别系统中的某些应用为例介绍该系统的使用情况。

图形识别在这里是指电路图、气象图、地理图、建筑图、工程图等。这一类图纸的一般识别过程如图 4(a) 流程所示。在这个流程中, 首先对图纸作预处理, 以消除由于各种原因造成的各种图纸缺陷。其中某些类型的噪声就可以用收缩处理和膨胀处理的交替使用予以消除。而矢量化过程则是把原来扫描得到的点阵图转换为矢量图, 以便于存储、查寻、编辑、修改和用绘图仪输出。有一种矢量化方法就是先对图形作细化处理, 得到代表原图形状的骨架, 再对该图形作跟踪, 矢量化处理。图 5 给出了用该系统对实际图形作的几种处理效果。图 5(a), (b) 分别为利用收缩和膨胀的组合使用, 消除原图中多余斑点(即噪声)和填补原图中的空白、间断的结果。其中(a)为先收缩后膨胀, (b)为先膨胀后收缩, (c)为细化结果。

图 4(b) 为以此功能硬件为基础的工程图自动处理系统, 由于采用了上述装置, 系统的处理速度是很高的。以细化为例, 如在处理过程中采用的时钟周期为 300ns, 扫描分辨率选用 300 点/英寸, 最大线宽为 1mm, 约 12 个像素宽, 则由一张 A0 幅面的图纸生成的文件约为 17MB, 需细化 6 次, 运算所用时间为分钟级。

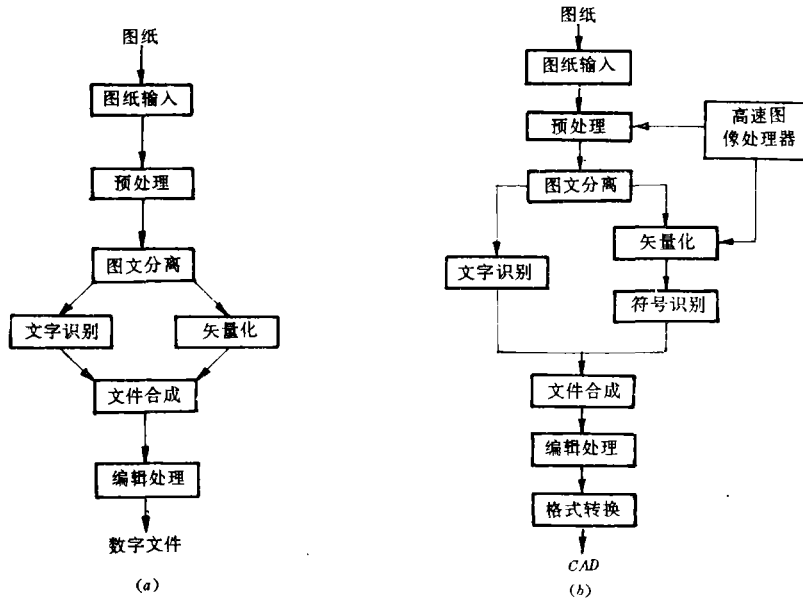


图 4 系统流程图

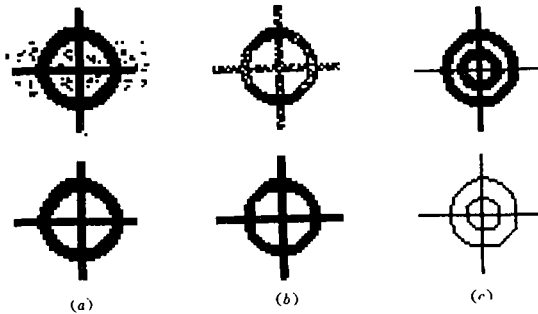


图 5 去噪及细化结果

5 结 论

本文介绍了一种高速多功能图像处理系统,该系统由硬件电路实现收缩、膨胀、轮廓跟踪和细化几种图像处理,具有保质、灵活、高速等特点。系统的处理速度只与硬件的延时有关,而与图形的复杂程度和主机无关。整个硬件电路可集中在一个标准的 PC 机接口板上,广泛用于各类图形处理操作中。

参 考 文 献

- [1] T. Pavlidis, Algorithms for Graphics and Image Processing, Comput. Sci. Rockville, MD, 1982
- [2] N. J. Naccache and R. Shinghal, SPTA, A proposed algorithm for thinning binary patterns, IEEE Trans, Systems Man Cybern, SMC-14, No. 3, 1984, 409-418
- [3] Y. S. Chen and W. H. Hsu, A modified fast parallel algorithm for thinning digital patterns, Pattern Recognition Letters, Vol, 7, No. 2, 1988, 99-106

[4] 田村秀行,《计算机图像处理技术》,北京师范大学出版社,1986

[5] Z. Y. Zhou, et al. , Transition from Hand Design to CAD, FICMT. 91, 1991

A High Speed Multifunction Image Processing System

Zhang Yi, Bing Wei and Zhou Zhaoying

(Dept. of Precision Instruments, Tsinghua University, Beijing 100084)

Abstract

A high speed multifunction binary image processing system is described in this paper. The system consists of logical circuits and can be used different image processing as contour tracking, thinning, dilating and eroding. The processing speed is limited only to the propagation delays of hardwares. An application example is also presented in the paper.

Key words: Binary image processing system, Logical circuits, Contour tracking