

一种快速二值图像欧拉数算法

姚 斌¹, 何立风^{1,2}, 康世英³, 赵 晓¹, 巢宇燕⁴

(1 陕西科技大学 电气与信息工程学院, 陕西 西安 710021;

2 日本爱知县立大学 信息科学学院, 日本 爱知县长久手市 480-1198;

3 咸阳师范学院 计算机学院, 陕西 咸阳 712000; 4 日本名古屋产业大学 环境商务信息学院,
日本 爱知县尾张旭市 488-8711)

摘 要: 欧拉数是二值图像的重要拓扑属性之一. 针对现有二值图像欧拉数算法中存在的像素重复检查问题, 提出了一种新的基于四方格计数方法的二值图像欧拉数算法, 定义了四方格的四种不同状态, 通过四方格之间的状态转换尽可能避免了像素重复检查, 同时利用数量较少的四方格实现二值图像欧拉数的计算以提高算法效率. 在不同类型图像上的实验结果表明, 该算法在大多数情况下都要优于其他现有的欧拉数算法.

关键词: 欧拉数; 图像处理; 模式识别; 拓扑属性; 计算机视觉

中图分类号: TP391.4

文献标识码: A

文章编号: 1000-7180(2017)07-0046-05

A Fast Algorithm for Computing the Euler Number in a Binary Image

YAO Bin¹, HE Li-feng^{1,2}, KANG Shi-ying³,
ZHAO Xiao¹, CHAO Yu-yan⁴

(1 College of Electrical and Information Engineering, Shaanxi Univ. of Science and Technology, Xian 710021, China; 2 Faculty of Information Science and Technology, Aichi Prefectural Univ., Aichi 480-1198, Japan; 3 School of Computer Science, Xianyang Normal Univ., Xianyang 712000, China; 4 Faculty of Environment, Information and Business Nagoya Sangyo Univ., Aichi 488-8711, Japan)

Abstract: The Euler number is one of the most important topological properties in a binary image. Aiming at the problem on repeated checking of pixels in the conventional computing algorithms, a new algorithm was proposed for computing the Euler number based on counting bit-quads in the given image. In the proposed algorithm, four states are constructed and taking advantage of transitions among the states, pixels can be avoided to check repeatedly. Furthermore, fewer bit-quad patterns were used in the proposed algorithm in order to improve the efficiency of the algorithm. Experimental results on various kinds of images demonstrated that the proposed algorithm is more efficient than conventional Euler number computing algorithms in most cases.

Key words: euler number; image processing; pattern recognition; topological property; computer vision

1 引言

二值图像(Binary Image)是指图像上的每一个像素只有两种可能的取值或灰度等级, 通常用黑白

图像或 B&W 图像表示. 二值图像经常用来描述文字或者图形, 其优点是在计算机中存储时占用空间少.

二值图像的欧拉数被定义为图像中包含的连通

收稿日期: 2016-10-20; 修回日期: 2016-11-17

基金项目: 国家自然科学基金青年基金项目(61603234); 国家自然科学基金面上项目(61471227); 陕西省教育厅科研计划项目(16JK1099)。

分量个数与孔洞数量之差,由于其在图像发生拉伸、收缩或旋转等变化时保持不变,有很强的鲁棒性,因此作为一种极其有用的图像特征被广泛应用于模式识别和计算机视觉领域,如医学图像处理^[1]、字符或数字识别^[2-3]、缺陷检测等。

在图像处理、尤其是实时图像处理过程中,由于图像中包含的数据较多,如何快速、准确地计算出图像的欧拉数显得尤为重要。到目前为止,已经有很多学者提出了不同类型的计算二值图像欧拉数的方法^[4-5]。文献[6]等人提出了利用二值图像一行中连续前景像素组成的图段数和相邻两行的邻接图段数计算欧拉数,这种算法对于密度较高和较低的二值图像较为有效;文献[7]给出了这种算法的另一种证明方式;文献[8]利用图论中的欧拉公式,简化了Gray等人提出的算法,实现了利用数量较少的四方格计算二值图像的欧拉数。为了进一步提高现有图像欧拉数算法的效率,本文提出了一种在文献[8]提出算法的基础上,根据Gray等人提出四方格计数的方法,同时利用四方格之间的状态转换减少像素重复检查次数以提高算法效率。实验结果表明,我们提出算法的性能在大多数情况下要优于现有算法。

为论述方便,我们用BT算法表示Gray提出的经典四方格计数算法,用RUN算法表示Bishnu等人提出的利用图段数和邻接图段数计算欧拉数的算法,用GT算法表示Yao等人提出的利用数量更少的四方格计算欧拉数的算法,用Ours表示本文提出的算法。此外,本文涉及到的算法都是基于像素间八邻接关系的。

2 现有算法概述

本文按照惯例,用“1”表示二值图像中的前景像素(目标像素),而用“0”表示背景像素(非目标像素)。

2.1 BT 算法

在BT算法中,为了计算二值图像的欧拉数,需要统计图像中包含的十种特定模式的 2×2 像素的四方格 P_1, P_2 和 P_3 数量,分别记为 N_1, N_2 和 N_3 ,如图1所示。

在算法实现过程中,只需要对图像从左到右、从上到下进行扫描,检查到某一个像素 $p(x, y)$ 时,只需要判断当前像素所处的四方格 $\begin{bmatrix} p(x, y) & p(x+1, y) \\ p(x, y+1) & p(x+1, y+1) \end{bmatrix}$ 是否属于需要统计的十种特定模式的四方格。如果是需要统计的,则相

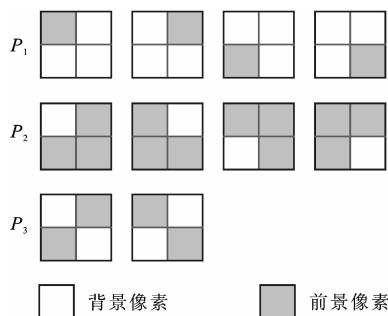


图1 BT算法中需要统计的四方格模式

应计数器累加,如果不需要统计,转而处理下一个像素。当图像中所有像素处理完毕,很容易可以得到要统计的三类、十种四方格的数量,根据式(1)就可以计算得到给定图像的欧拉数。不难看出,如果图像分辨率为 $M \times N$ 像素,则BT算法总的像素读取次数为 $4 \times M \times N$ 。

$$E = (N_1 - N_2 - 2N_3) / 4 \quad (1)$$

2.2 RUN 算法

2005年,Bishnu等人提出利用二值图像中的图段数和邻接图段数来计算欧拉数的算法。图段是由在图像某一行中连续的前景像素构成的,一行中可以有多个图段。图像中图段数 R 指的是图像所有行的图段数的总和。如果相邻两行的不同图段中存在两个前景像素构成八邻接关系,则构成一个邻接图段。图像中的邻接图段数 NR 指的是图像中所有邻接图段个数的总和。如图2所示,第一行中有3个图段,第二行中有3个图段,总的图段数为6;相邻图段数为4(用椭圆标记)。这种算法需要对图像进行逐行扫描,获取每一行的图段数和相邻行的邻接图段数。当图像扫描完毕,可以得到图像中的图段数和邻接图段数,用式(2)就可以得到给定图像的欧拉数。

$$E = R - NR \quad (2)$$

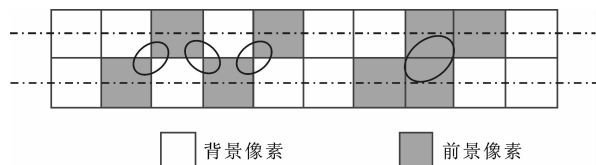


图2 RUN算法中的图段和邻接图段示例

在图2中,图像的欧拉数 $E = 3 + 3 - 4 = 2$ 。RUN算法中,在对图像进行逐行扫描时,除了要计算当前行的图段个数,还需要判断、统计相邻行的邻接图段数,计算过程比BT算法稍微复杂一些,但像素读取次数相对少一些。

2.3 GT 算法

用 GT 算法计算二值图像的欧拉数,首先需要将给定图像图形化,即将每一个前景像素当作一个结点,如果相邻的两个结点相互形成八邻接关系,则在两个结点间添加一条边,这样就得到要最终用来计算欧拉数的图形。

欧拉定理:设 G 是平面图,有 c 个连通分支, v 个结点, e 条边, r 个面, 则公式 $v - e + r = c + 1$ 成立。

通过对图像分析易知,欧拉定理中的连通分支数即图像中的连通分量数,而“面”包括基本面(s)、孔洞(h)和图形外侧的一个无限面,故面数 $r = s + h + 1$ 。将 r 代入欧拉定理,可得 $v - e + (s + h + 1) = c + 1$, 整理后可得 $c - h = v - e + s$, 于是根据欧拉数的定义可知 $E = c - h = v - e + s$ 。这里需要注意的是八邻接欧拉数算法中的基本面为腰长为 1 的等腰直角三角形。当结点数、边数和基本面数都被计算出来,用公式 $E = v - e + s$ 就可以得到给定图像的欧拉数。

为了计算图像对应图形中的结点数 v , 边数 e 和基本面数 s 。按照文献[8]的分析,只需要对图像进行逐行扫描,统计图像中包含的三种特定四方格模式 P_1, P_2 和 P_3 就可以了,如图 3 所示。图像扫描方法和 BT 算法类似,当图像扫描完毕,统计出三种四方格模式的数量,记为 N_1, N_2 和 N_3 , 利用公式(3)就可以得到给定图像的欧拉数。

$$E = N_1 - N_2 - N_3 \tag{3}$$

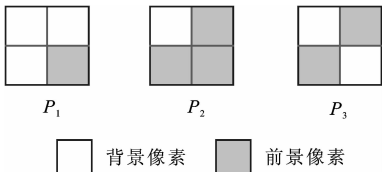


图 3 GT 算法中需要统计的四方格模式

3 改进的算法

文献[8]给出的算法中,尽管在扫描图像的过程中需要统计的四方格数量由十种减少到三种,但是对于每一个像素来说,仍然需要检查其所在四方格的四个像素来判断当前四方格是否需要统计的。也就是说,对于一个 $M \times N$ 像素的图像来说,总的像素读取次数仍然是 $4 \times M \times N$ 。

然而,这种算法存在像素被重复检查的问题。如图 4 所示,当处理当前像素 P_1 时,我们需要检查像素 P_1 及其所在四方格的其他三个像素 P_2, P_6 和

P_7 ,也就是四方格 $\begin{bmatrix} P_1 & P_2 \\ P_6 & P_7 \end{bmatrix}$ 。接下来处理像素 P_2 时,我们需要检查像素 P_2 及其所在四方格的其他三个像素 P_3, P_7 和 P_8 ,也就是四方格 $\begin{bmatrix} P_2 & P_3 \\ P_7 & P_8 \end{bmatrix}$ 。很明显,像素 P_2 和 P_7 被重复检查了。也就是说,在处理某一个四方格 $\begin{bmatrix} p(x,y) & p(x+1,y) \\ p(x,y+1) & p(x+1,y+1) \end{bmatrix}$ 时,像素 $\begin{bmatrix} p(x,y) \\ p(x,y+1) \end{bmatrix}$ 已经在处理上一个像素时检查过了,我们只需要检查像素 $\begin{bmatrix} p(x+1,y) \\ p(x+1,y+1) \end{bmatrix}$ 就可以得到当前四方格的类型。

P_1	P_2	P_3	P_4	P_5
P_6	P_7	P_8	P_9	P_{10}
P_{11}	P_{12}	P_{13}	P_{14}	P_{15}
P_{16}	P_{17}	P_{18}	P_{19}	P_{20}

图 4 现有基于四方格的算法中存在问题示例

为了避免上述像素检查方法造成的像素重复检查,我们提出利用四方格状态转换来实现。为此,需要保存像素 $\begin{bmatrix} p(x,y) \\ p(x,y+1) \end{bmatrix}$ 的状态。由于在二值图像中像素值只有 $[0, 1]$ 两种,因此我们需要定义四种状态 $St_1 - St_4$, 分别对应像素 $p(x, y)$ 和 $p(x, y + 1)$ 的值为 $\langle 0, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 0 \rangle$ 和 $\langle 1, 1 \rangle$, 如图 5 所示,在实际图像处理过程中,我们处理每一个像素时,只需要检查像素 P_x 和 P_y 。

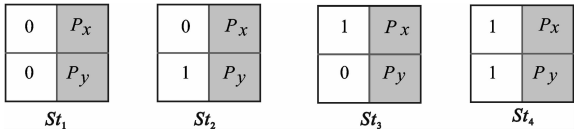


图 5 提出的算法中定义的状态

通常情况下,由于待处理图像的边缘是背景像素,所以处理从状态 St_1 开始。如图 3 所示,由于需要计数的四方格类型有三种,我们事先设置三个计数器 C_1, C_2 和 C_3 , 分别对应需要统计的四方格模式 P_1, P_2 和 P_3 的个数。

(1) 如果像素 P_x 和 P_y 的值为 $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$, 我们知道当前四方格为 $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$, 这种类型的四方格并不需要统

计,因此我们接下来处理下一像素,这时只需要跳转到状态 St_1 ,下一像素的处理仍然从状态 St_1 开始.

(2) 如果像素 P_x 和 P_y 的值为 $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$,我们知道当前四方格为 $\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$,如图3所示,这种类型的四方格是需要计数的四方格 P_1 ,对应计数器 C_1 加1. 接下来我们处理下一像素,由于像素 P_x 和 P_y 的值为 $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$,此时我们需要将状态转到 St_2 ,也就是说处理下一像素时从 St_2 开始.

(3) 如果像素 P_x 和 P_y 的值为 $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$,我们知道当前四方格为 $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$,这种类型的四方格同样不需要统计,接下来处理下一像素,由于像素 P_x 和 P_y 的值为 $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$,此时我们需要将状态转到 St_3 ,处理下一像素时从 St_3 开始.

(4) 如果像素 P_x 和 P_y 的值为 $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$,我们知道当前四方格为 $\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$,这种类型的四方格同样不需要统计,接下来处理下一像素,由于像素 P_x 和 P_y 的值为 $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$,此时我们需要将状态转到 St_4 ,处理下一像素时从 St_4 开始.

在后续的处理过程中,如果当前状态是 St_2 ,我们同样需要检查像素 P_x 和 P_y .

(1) 如果像素 P_x 和 P_y 的值为 $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$,我们知道当前四方格为 $\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$,并不需要统计,接下来处理下一像素,只需要跳转到状态 St_1 .

(2) 如果像素 P_x 和 P_y 的值为 $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$,我们知道当前四方格为 $\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$,并不需要统计,接下来处理下一像素,只需要跳转到状态 St_2 .

(3) 如果像素 P_x 和 P_y 的值为 $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$,我们知道当前四方格为 $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$,是需要计数的四方格 P_3 ,对应计数器 C_3 加1,接下来处理下一像素,只需要跳转到

状态 St_3 .

(4) 如果像素 P_x 和 P_y 的值为 $\begin{bmatrix} 1 \\ 1 \end{bmatrix}$,我们知道当前四方格为 $\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$,是需要计数的四方格 P_2 ,对应计数器 C_2 加1,接下来处理下一像素,只需要跳转到状态 St_4 .

对于状态 St_3 和 St_4 ,可以类似地按照上述方法处理. 当所有像素处理完毕,我们可以得到四方格 P_1, P_2 和 P_3 的数量 C_1, C_2 和 C_3 ,利用公式(3) 很容易得到给定图像的欧拉数.

4 实验结果

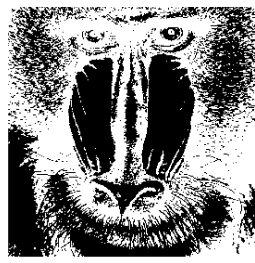
为了测试提出的算法的正确性和有效性,我们采用四种不同类型的图像(自然图像、医学图像、纹理图像和人工图像)对算法进行测试,并把我们的运行结果与 BT 算法、RUN 算法及 GT 算法进行比较.

实验测试图像包括下载自加利福尼亚大学、东京大学和哥伦比亚大学等国外高校图像数据库的50幅自然图像、7幅纹理图像、25幅医学图像及4幅人工图像(如棋盘状、螺旋状等). 所有测试图像分辨率均为 512×512 像素. 测试平台为台式机(Intel Core i5-3470 CPU@3.20 GHz, 4 GB 内存,操作系统 Ubuntu Linux),编译器为 GNU C compiler(版本 4.6.1). 为保证实验结果的准确性,所有计算图像的欧拉数的程序均被运行 5 000 次,取平均值作为实验结果.

从测试结果可以看出,对于所有图像,我们提出算法的测试结果与另外三种算法测试结果一致. 图6给出了部分测试图像(一幅自然图像和一幅医学图像)的测试结果,包括欧拉数和各种算法执行时间(ms).



Euler number: -347
BT: 1.43 RUN: 1.25
GT: 1.15 Ours: 0.80



Euler number: -1 286
BT: 1.84 RUN: 1.69
GT: 1.47 Ours: 1.19



Euler number: -126 Euler number: 31
BT: 1.27 RUN: 0.91 BT: 1.27 RUN: 0.76
GT: 1.02 Ours: 0.64 GT: 1.01 Ours: 0.60

图 6 部分实验图像测试结果

四种算法在不同类型图像的最长、最短和平均执行时间的测试结果如表 1 所示. 从表 1 可以看出, 对于所有的自然图像、医学图像和纹理图像, 在计算欧拉数时, 不论是最长时间 (Max.)、最短时间 (Min.) 还是平均时间 (Mean), 本文算法的执行时间的都小于 BT 算法、RUN 算法和 GT 算法.

表 1 四种算法在测试不同类型图像时算法的最长、最短以及平均执行时间

算法		BT	RUN	GT	Ours	ms
自然图像	Max.	1.86	1.69	1.50	1.20	
	Mean	1.42	1.07	1.14	0.77	
	Min.	1.10	0.61	0.89	0.51	
医学图像	Max.	1.47	1.07	1.18	0.80	
	Mean	1.29	0.92	1.03	0.65	
	Min.	1.17	0.75	0.96	0.59	
纹理图像	Max.	1.73	1.66	1.35	1.04	
	Mean	1.38	1.35	1.09	0.77	
	Min.	1.00	1.04	0.83	0.51	
人工图像	Max.	1.11	1.03	0.86	0.50	
	Mean	0.70	0.67	0.53	0.32	
	Min.	0.27	0.24	0.22	0.13	

5 结束语

为了解决二值图像欧拉数的快速计算问题, 本文提出了一种新的二值图像欧拉数的算法. 在本文的算法中, 利用数量更少的四方格模式来计算, 同时, 应用被处理的四方格间的状态转换, 大大减少了像素重复检查的次数, 将算法中总的像素读取次数降低为图像像素数量的两倍. 实验结果表明, 本文的算法对被测的四种类型图像的欧拉数计算结果均与 BT 算法、RUN 算法及 GT 算法得到的结果一致,

而且对于所有被测图像来说, 本文算法的性能优于现有的 BT 算法、RUN 算法及 GT 算法. 在算法实现过程中, 我们发现在对图像进行逐行扫描、检查每个四方格来计算图像欧拉数时, 在图像的垂直方向上仍然存在像素重复检查的现象. 所以在接下来的工作中, 笔者将对算法进行优化和改进, 以期得到性能更好的二值图像欧拉数算法.

参考文献:

[1] Pogue B W, Mycek M A, Harper D. Image analysis for discrimination of cervical neoplasia[J]. Journal of Biomedical Optics, 2000, 5(1):72-82.

[2] 毕记省, 田璐, 张广有. 基于认知理论的地图点状符号智能识别[J]. 海洋测绘, 2016, 36(1):65-67.

[3] 陈玮, 曹志广, 李剑平. 改进的模板匹配方法在车牌识别中的应用[J]. 计算机工程与设计, 2013, 34(5): 1808-1811.

[4] Gray S B. Local properties of binary images in two dimensions[J]. IEEE Trans. Computers, 1971, C-20: 551-561.

[5] Juan L D S, Azuela J H S. On the computation of the Euler number of a binary object[J]. Pattern Recognition, 1996, 29(3): 471-476.

[6] Bishnu A, Bhattachary B B, Kundu M K, et al. A pipeline architecture for computing the Euler number of a binary image[J]. Journal of Systems Architecture, 2005, 51(8): 470-487.

[7] 林小竹, 吴艺娟, 傅旦丹. 图像欧拉数计算复杂度分析[J]. 微电子学与计算机, 2008, 25(7):14-16.

[8] Yao B, He L, Kang S, et al. A novel bit-quad-based Euler number computing algorithm[J]. Springerplus, 2015, 4(1):1-16.

作者简介:

姚 斌 男, (1981-), 博士研究生, 讲师. 研究方向为图像处理、图像特征提取. E-mail: yaobin992@126.com.

何立风 男, (1963-), 博士, 教授. 研究方向为模式识别、图像处理.

康世英 女, (1980-), 硕士, 讲师. 研究方向为计算机视觉、图像处理.

赵 晓 女, (1978-), 博士研究生, 讲师. 研究方向为图像处理、图像特征提取.

巢宇燕 女, (1963-), 博士, 教授. 研究方向为人工智能、图像处理.