

# 一种规则几何图形的计算机识别方法

孙来军, 李江游, 候影, 叶光忠

(黑龙江大学 黑龙江省电子工程高校重点实验室, 黑龙江 哈尔滨 150080)

**摘要:** 许多领域在进行图片处理时都需要识别出图片中的几何图形, 传统的霍夫变换、不变矩等识别方法都存在着耗费大量系统内存、计算速度慢等缺点。基于此, 提出一种新的几何图形的识别方法。介绍了该方法的实现原理和实现过程。最后利用 C++Builder 编写软件对这种方法进行了测试, 证明了该方法的正确性和可靠性。

**关键词:** 几何图形; 识别; 像素

中图分类号: TP751

文献标识码: A

文章编号: 1674-7720(2011)09-0042-04

## A computer recognition method of regular geometry graphics

Sun Laijun, Li Jiangyou, Hou Ying, Ye Guangzhong

(Key Laboratory of Electronics Engineering College of Heilongjiang Province, Heilongjiang University, Harbin 150080, China)

**Abstract:** It is necessary to identify the geometric shapes in the picture processing in many fields. Some of the traditional identification methods such as the Hough transform has a great amount of system memory cost, calculation speed slow shortcomings. So this paper presents a new method of the identification of geometric shapes, and discusses the method of principle and the realization of the process in detail. Finally, use C++ Builder programs a software to test this method, it proves the validity and reliability of the method.

**Key words:** geometric figure; recognition; pixel

在计算机视觉领域中, 如何从当前的图像中提取出所需要的特征信息是图像识别的关键所在。在许多应用场合中需要快速准确地识别出图像中的几何图形(如圆、矩形、三角形等), 从而进行更深一步的操作。例如: 在基于图像处理的自动报靶系统中, 对靶环的识别就属于这一类应用<sup>[1]</sup>。在这些应用领域, 许多学者提出了多种方法, 如霍夫变换、不变矩、骨架构造识别等。这些传统的识别方法虽然检测的准确性很高而且算法成熟, 但在进行几何图形识别时都存在需要耗费大量系统内存、计算速度较慢等缺点, 并不适合要求对几何图形进行快速检测的领域<sup>[2-3]</sup>。因此, 本文提出了一种全新的基于几何图形(周长、面积以及顶点个数)的图形识别方法。经过测试, 该方法可以快速有效地检测出一幅图片中的几何图形并给出该图形的周长、面积等参数。

### 1 图形识别的基本原理

一般情况下, 从客观景物得到的图像是二维的, 可以用一个二维数组  $f(x, y)$  来表示。  $x, y$  表示二维空间中

一个坐标点的位置, 而  $f(x, y)$  则是代表图像在点  $(x, y)$  的某种性质的数值。为了能用计算机对图像进行加工, 需要把连续的图像在坐标空间  $(x, y)$  和性质空间  $F$  都离散化, 这种离散化的图像是数字图像, 可以用  $I(x, c)$  来表示。  $I(x, c)$  代表离散后的  $f(x, y)$ 。为了方便, 本方法仍用  $f(x, y)$  代表数字图像, 图像中每个基本单元称为图像元素(简称像素)。一幅图像必须在空间和灰度上都离散化才能被计算机处理。空间坐标的离散化称为空间采样。假设用一个  $m \times n$  维的数组中等距地采样来近似表示一幅连续图像  $f(x, y)$ , 即<sup>[4]</sup>:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,m-1) \\ f(1,0) & f(1,1) & \cdots & f(1,m-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(n-1,0) & f(n-1,1) & \cdots & f(n-1,m-1) \end{bmatrix}$$

式中, 每个元素都是一个离散变量, 其右边的矩阵代表数字图像, 数组的每个元素即对应像素。由此可见, 对图片的处理实际上是对组成图片的像素进行操作。在

实际的检测中,一般是根据几何图形所在区域的像素点的灰度级别与图片背景的像素点的灰度级别存在的差异,将属于几何图形的像素点检测出来。对于规则几何图形的识别,本文所提出的算法是根据几何图形的周长、面积以及顶点个数来判断。下面介绍圆形、矩形(其他普通四边形也适用)、三角形的识别原理。

(1)圆形:圆形是所有规则几何图形中最特殊的一种图形,无论是求它的周长还是面积都要用到 $\pi$ 。虽然 $\pi$ 是一个无穷不重复循环小数,但可以把它看作一个常数,因此,可以利用这一点来确定所识别的图形为圆形。假设任意一个圆形,已知半径为 $r$ ,周长为 $L$ ,面积为 $S$ ,根据计算公式: $L=2\times\pi\times r$ , $S=\pi\times r^2$ ,则:

$$\text{ratio}=L^2/S=(2\times\pi\times r)^2/\pi\times r^2=4\pi$$

由计算式不难看出,半径 $r$ 对比值没有任何影响,也就是说无论圆形面积有多大,其周长的平方与面积的比为一个定值。但需要指出的是:实际用来检测的图片中的圆形并不一定是标准的圆形,因此,其周长的平方与面积的比也不是绝对等于 $4\pi$ 。不过这个比值会在一个固定的范围内,所以只要所识别的图形满足这个条件即为圆形。

(2)矩形:矩形虽然也是日常生活中比较常见的一种特殊几何图形,但并不能用上述的方法对其进行识别。假设任意一个矩形,已知其长和宽分别为 $l$ 和 $w$ ,周长为 $L$ ,面积为 $S$ ,根据计算公式: $L=2\times(l+w)$ , $S=l\times w$ ,则:

$$\text{ratio}=L^2/S=4\times(l+w)^2/l\times w=4\times(l/w+w/l+2)\geq 4\times(2\times\sqrt{l/w\times w/l}+2)=16。$$

由灰度比值可知,该比值只有一个确定的下限,而无确定的上限,因此,不能利用周长的平方与面积的比对矩形进行识别。但矩形也有其特殊的地方,即它有4个顶点,而且只有4个顶点。只要对所检测的图形的边界点进行判断并计算顶点的个数,如果个数等于4,则为矩形(或其他四边形);否则就为其他几何图形。与圆形的识别相类似,在对矩形的实际检测过程中也可能会出现顶点个数不为4的现象,这是由于图形绘制不规范所造成的。但只要对已经检测出来的顶点再进行一次“同邻域”判断,看其中是否有些顶点属于同一个邻域范围(一般为8邻域),如果有,只留下一个而将其他的去除,这样就可以对结果进行很好的修正,从而可避免错误检测的出现。利用这种方法就可以很好地识别出所检测的图形为矩形。

(3)三角形:三角形的识别与矩形的识别方法相类似,也是利用顶点的个数来判断图形所属的几何形状,这里不再赘述。

## 2 图形识别的实现过程

利用计算机进行图形识别的过程如图1所示。

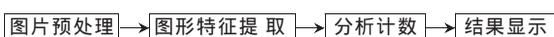


图1 图形识别过程

### 2.1 预处理

从图1可以看出,对一幅图片进行操作之前需进行必要的预处理,这是因为图片在绘制过程中或输入到计算机的过程中可能引入噪声。通常的预处理过程包括灰度化处理、二值化处理、平滑、滤波等。其步骤如下:

(1)灰度化:图片实际上是一个像素的二维数组,其中每个像素的颜色由构成这种颜色的红(R)、绿(G)、蓝(B)三种基本色的分量来表示。以真彩图片为例,典型情况下,颜色的每个组分都用一个字节来表示,因此为每种颜色组分提供了256个级别。这种方法为每个像素使用3B,允许达到 $256\times 256\times 256=16\,777\,216$ 或1600万种RGB值(颜色)。由于彩色信息的区分度不高,如果直接对彩色图片的像素进行操作,所得到的检测结果精度也不高。而将图片进行灰度处理后,图片以亮度信息为主要信息,彩色信息为次要信息,则可以在很大程度上提高检测精度,因此需将图片进行灰度处理,把彩色图片转换为灰度图。灰度图处理采用的方法有:最大值转换法、平均值转换法、黄金权重转换法等,本文在进行灰度处理时采用的是最大值(亮度)转换法。经过这种方法处理后的图片每一个像素的RGB值都是相同的,即RGB值的范围从(0,0,0)、(1,1,1)一直到(255,255,255)。其中,(0,0,0)是全黑色,(255,255,255)是全白色,中间的是灰色<sup>[5]</sup>。实现这个过程的部分源代码为:

```
//利用 GetxValue 取出每一个点的像素的 RGB 值
r=GetRValue(color1);
g=GetGValue(color1);
b=GetBValue(color1);
//找出 RGB 值中最大的值
np=r>g? r:g;
np=np>b? np:b;
//将图片变为亮度最高的灰度图
Image1->Picture->Bitmap->Canvas->Pixels[i][j]=
TColor(RGB(np, np, np));
```

(2)二值化:虽然经过灰度处理后图片的主要信息仅为亮度信息,但有效检测区域的像素点的灰度级别并不统一。因此,灰度处理后往往还需要对图片再进行二值化处理,以使所要检测的几何图形的区域变为较深的颜色,背景的部分变为较浅的颜色。这样可以更进一步加强区分度,提高检测的精度<sup>[6]</sup>。本文所使用的二值化方法为大律法,大律法又称最大类间方差法,其处理原理为:取某个灰度值,以它为分界点将图像分为灰度值大小的两类,分别计算这两类中的像素点数及灰度平均值;然后,计算它们的类间方差;最后取所有灰度的类间方差中的最大值对应的灰度为阈值。类间方差的计算公式如下:

$$w(i)=n_1(i)n_2(i)(v_1(i)-v_2(i))^2$$

其中, $n_1(i)$ 、 $n_2(i)$ 分别为灰度小于 $i$ 的像素以及大于等于 $i$ 的像素的个数, $v_1(i)$ 、 $v_2(i)$ 分别为灰度的平均值。

实现这个过程的部分源代码为:

```
//二值化处理
for(i=0;i<w;i++)
{
for(j=0;j<h;j++)
{
color=Image1->Picture->Bitmap->Canvas->
Pixels[i][j];
ave=Dalu(gray);
if(GetRValue(color)>ave)
//将背景点变为白色
Image1->Picture->Bitmap->Canvas->Pixels[i][j]=
TColor(255,255,255);
Else
//将几何图形区域变为黑色
Image1->Picture->Bitmap->Canvas->Pixels[i][j]=
TColor(0,0,0);
}
}
}
```

(3)平滑和滤波。在进行完步骤(1)、(2)的预处理后,如果几何图形还是不十分明显,可以用高斯平滑滤波器和拉普拉斯锐化滤波器结合计算即可得到几何图形突出的图片<sup>[7]</sup>。

## 2.2 特征提取

完成图片的预处理后,即可进行图片的特征提取。由于是规则的几何图形,因此所要提取的特征信息即为几何图形的周长、面积和顶点的个数。计算图形周长和顶点个数时用的是 Laplace 边缘检测算法。Laplace 算法有两类:正相 Laplace 算法和反相 Laplace 算法,两者都反映了一类边缘,但其所用模板和参数不同<sup>[8]</sup>。假设一个 3×3 邻域内的像素分布如图 2 所示,设 LP、LN 分别为正相 Laplace 算法和反相 Laplace 算法的处理结果,其计算公式如下:

$$LP=[f(A0)+f(A1)+\dots+f(A7)]-8f(C)$$

$$LN=8f(C)-[f(A0)+f(A1)+\dots+f(A7)]$$

这两个公式也可用卷积模板来表示,如图 3(b)、(c)所示。由图可知,边缘检测模板中各因子之和为 0,其物理意义是:在图像的均匀区域处理输出为 0,在像素数据有突变的地方则输出较大的值。以此来检测图像中的边缘。

A0	A1	A2
A7	C	A3
A6	A5	A4

图 2 3×3 邻域内像素分布

+1	+1	+1
+1	-8	+1
+1	+1	+1

(a)正相模板

-1	-1	-1
-1	+8	-1
-1	-1	-1

(b)反相模板

图 3 Laplace 边缘检测模板

用 Laplace 边缘检测算法对几何图形进行检测完成后,累加边界点的个数即为几何图形的周长。然后再对这些边界点进行进一步区分,找出顶点。检测几何图形

的面积则要简单得多,只要统计出有效检测区域内所有符合条件的点即为几何图形的面积。实现这个过程的部分源代码为:

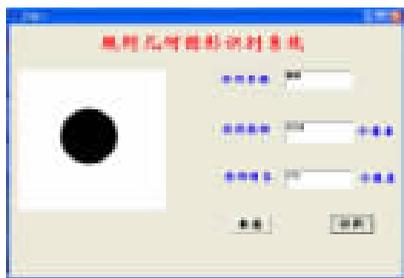
```
//计算周长、面积和顶点
int x,y,sum=0;
for(i=0;i<w;i++)
{
for(j=0;j<h;j++)
{
color=Image1->Picture->Bitmap->Canvas->
Pixels[i][j];
if(GetRValue(color)==0)
{
mianji++; //计算面积
if(Laplace(GetRValue(color)))
zhouchang++; //计算周长
if(Dingdian(GetRValue(color)))
dingdian++; //计算顶点
}
}
}
}
```

## 2.3 分析判断

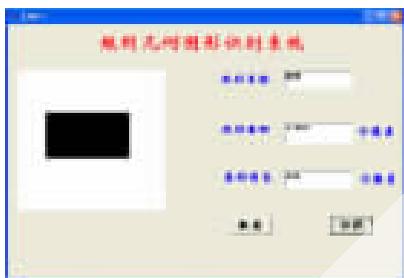
在完成图像的预处理和特征提取之后,便可以开始根据特征信息对几何图形的形状进行判断。在这个过程中首先判断几何图形是否为圆形,即判断这个图形周长的平方与面积的比是否在规定的范围内。如果在,则为圆形;否则,再判断顶点的个数,等于 3 即为三角形,等于 4 即为矩形。在识别几何图形形状的同时还可输出这个图形的周长和面积。因为规则几何图形的周长、面积和顶点坐标已知,故可计算出边长、形心等参数。实现这个过程的部分源代码为:

```
//根据周长的平方与面积的比以及顶点的个数确定图形的形状
float ratio;
ratio=(zhouchang*1.0*zhouchang*1.0)/mianji*1.0;
//判断是否为圆形
if(ratio>=9&&ratio<=15)
Edit1->Text="圆形或椭圆";
else
{
//判断是否为三角形
if(dingdian==3)
Edit1->Text="三角形";
//判断是否为矩形
if(dingdian==4)
Edit1->Text="矩形";
}
//输出面积和周长
Edit2->Text=mianji;
Edit3->Text=zhouchang;
基于上述的基本原理和实现过程,利用 C++
```

Builder6.0 编写软件, 几何图形识别结果如图 4 所示<sup>[9]</sup>。从识别结果可知, 本文所述的方法可以快速而准确地识别出图像中的规则几何图形, 从而证明了该法的可行性和正确性。



(a)圆形



(b)矩形



(c)三角形

图 4 几何图形识别结果

与传统的检测方法相比, 本文提出的基于几何图形周长、面积和顶点个数的检测方法可以很好地识别出几

何图形的形状, 并能给出边长、形心等其他参数, 且识别速度快、准确率高。但这种识别方法也存在一些不足, 它的识别对象只是几种简单的规则几何图形, 无法识别复杂的不规则几何图形。这些不足还有待于在日后的研究中不断克服, 以使这种检测方法能够更加稳定实用。

#### 参考文献

- [1] 顾筠. 基于霍夫变换和几何特性的图像识别方法[J]. 自动化博览, 2002, 8(1): 86-88.
- [2] 邱桑敏. 一种快速霍夫变换算法[J]. 计算机工程, 2004, 1(2): 148-150.
- [3] 李海军, 宁玉富. 规则几何图形的识别分析与算法实现[J]. 常熟理工学院学报, 2007, 4(4): 111-115.
- [4] 董莉, 王玲, 江沸波. 一种基于医学图像中感兴趣区域的数字水印算法[J]. 电子技术应用, 2007, 33(1): 136-139.
- [5] 高士忠, 卢易枫, 蔡振伟, 等. 基于灰度共生矩阵的帘子布疵点检测研究[J]. 电子技术应用, 2008, 34(1): 117-119.
- [6] 刘玉红, 熊兴良. 图像的带参二值化理论和技术及其应用[J]. 计算机应用研究, 2008, 25(1): 277-280.
- [7] 鲁志波, 王鑫, 胡国恩. 结合结构张量和 Wiener 滤波的图像去噪算法[J]. 计算机科学, 2007, 34(7): 229-231.
- [8] 刘佳, 肖晓明, 彭骏驰. 基于改进的 Laplacian 算子的图像边缘检测[J]. 电子技术应用, 2006, 32(11): 31-32.
- [9] 蒙祖强, 龚涛. C++ Builder 程序员成长攻略[M]. 北京: 中国水利水电出版社, 2007.

(收稿日期: 2010-12-15)

#### 作者简介:

孙来军, 男, 1977 年生, 博士, 副教授, 主要研究方向: 状态检测与故障诊断, 无损检测技术。

李江游, 男, 1988 年生, 学士, 主要研究方向: 智能仪器与智能测控技术。

侯影, 女, 1987 年生, 学士, 主要研究方向: 智能仪器与智能测控技术。