

数字图像处理基础知识

左 力

一. 数字图像处理的定义

图像处理：对图像进行各种加工以改善图像的效果，为图像分析做准备。它是图像到图像的过程。

点运算：灰度变换、阈值变换、灰度均衡等

几何变换：移动、旋转、扭曲校正等。

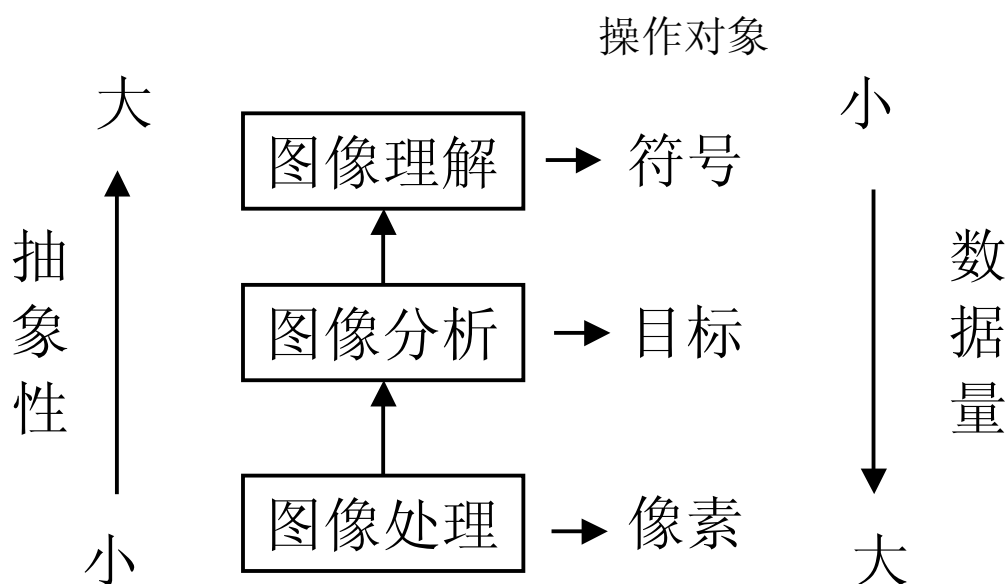
图像分析：对图像中感兴趣的目标进行检测和测量(特征提取)，以获得客观的信息。它是图像到数据的过程。

边缘检测与提取、图像分割、几何测量、模板匹配等。

图像理解：研究图像中各目标的性质和其相互关系，理解图像的含义。

字符识别（OCR）、产品质量检验（目检）、人脸识别、自动驾驶、医学图像和地貌图像的自动判读理解等。

图像处理、图像分析和图像理解的关系：



图像工程：

- s 是图像理论、图像技术及其应用的新学科。
- s 数学、物理学、生理学、心理学、计算机科学是其基础；
- s 得到人工智能、神经网络、模糊逻辑、遗传算法等新理论与新技术的支持；
- s 和模式识别、计算机视觉、计算机图形学等专业相互交叉；
- s 其发展应用与工业自动化、医学、遥感、信息等领域不可分割。

二. 图像的采集、格式

图像处理系统硬件结构:

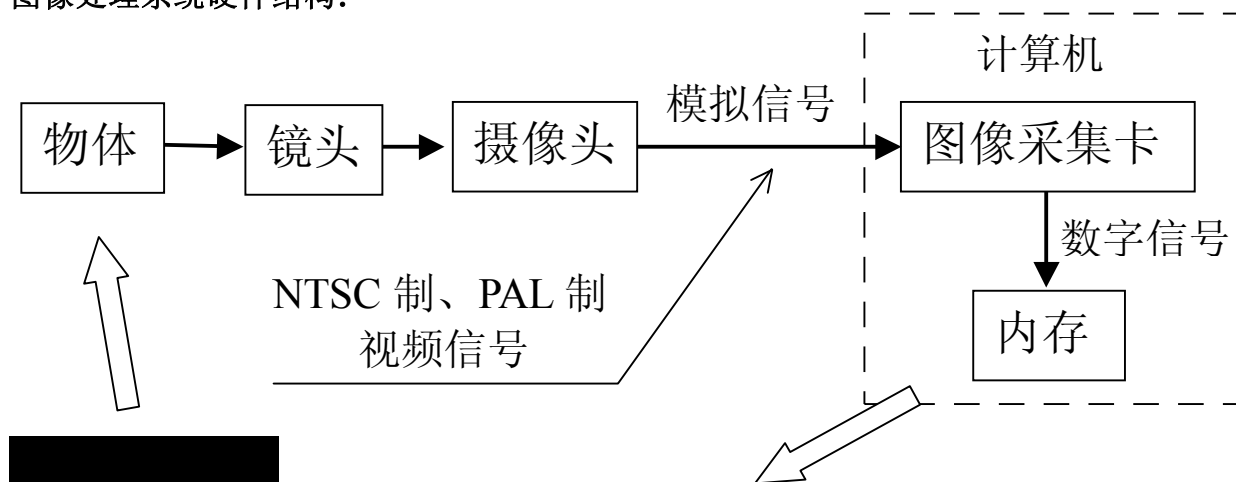


图 1

像素矩阵

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	255	255	255	255	0	0
0	128	128	128	128	128	128	0
0	128	128	128	128	128	128	0
0	128	128	128	128	128	128	0
0	128	128	128	128	128	128	0
0	0	0	0	0	0	0	0

行

0

列

表 1

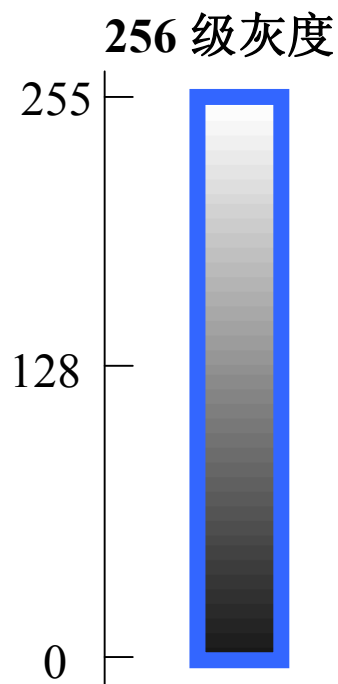


图 2

图像处理系统一般使用 256 级灰度图像，即 8 位黑白图像，其 1 个像素由 1 个字节描述。0 表示黑色，255 为白色；其它中间灰度见图 2。

一个立方形物体的照片如图 1 所示。通过图像采集卡后，其像素矩阵如表 1 所示。

图像的大小、分辨率和测量精度有关

NTSC 制: 768 × 576 像素

PAL 制: 640 × 480 像素

注：一个 8 位二进制数为 1 个字节。其最小值为 0，最大值为 255。

三. 阈值分割、形心计算

图 3.1 为一园形物体照片，下面介绍将物体与背景分割开来的方法，和计算园形物体形心及半径的方法。



图 3.1 一园形物体的图像

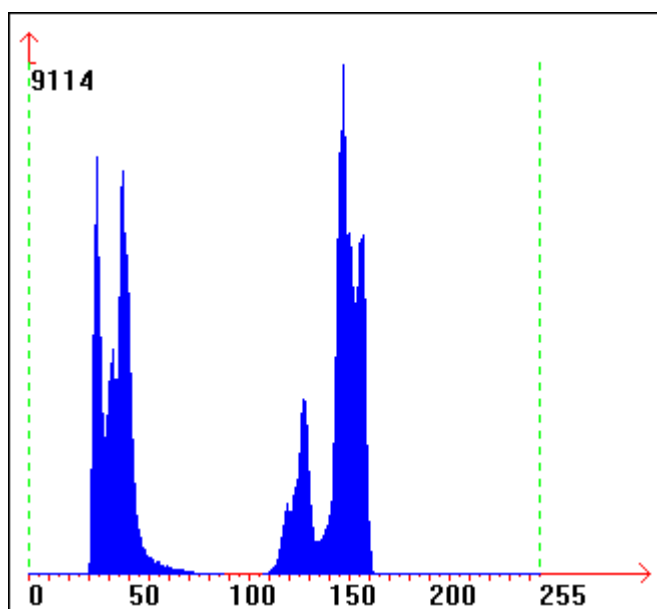


图 3.2 图 3.1 的灰度直方图（线性坐标）

灰度直方图是数字图像处理中一个最简单、最有用的工具。它表示图象中具有每种灰度级的像素的个数，反映图像中每种灰度出现的频率。如图 3.2 所示，灰度直方图的横坐标是灰度级，纵坐标是该灰度级出现的频率，它是图像的最基本的统计特征。

最简单、有效的方法就是取双峰间谷底的灰度值为阈值，根据该阈值对图像进行分割，将物体与背景分离开来，然后再进行其它处理。

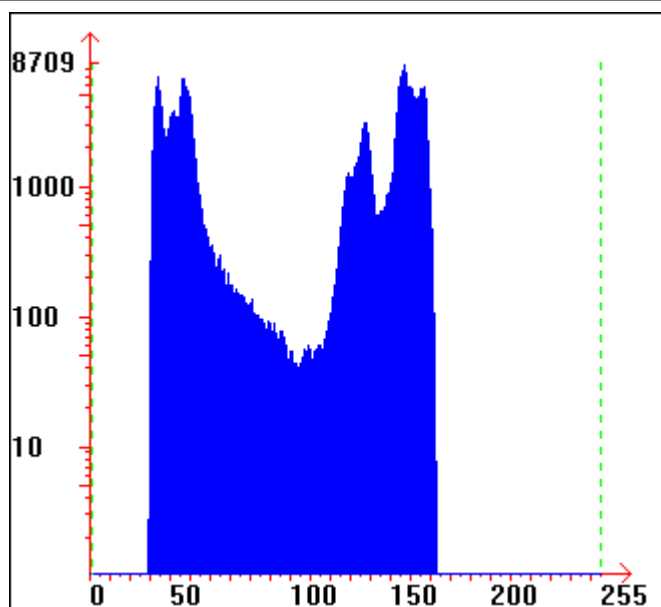


图 3.3 对数坐标直方图，平滑前

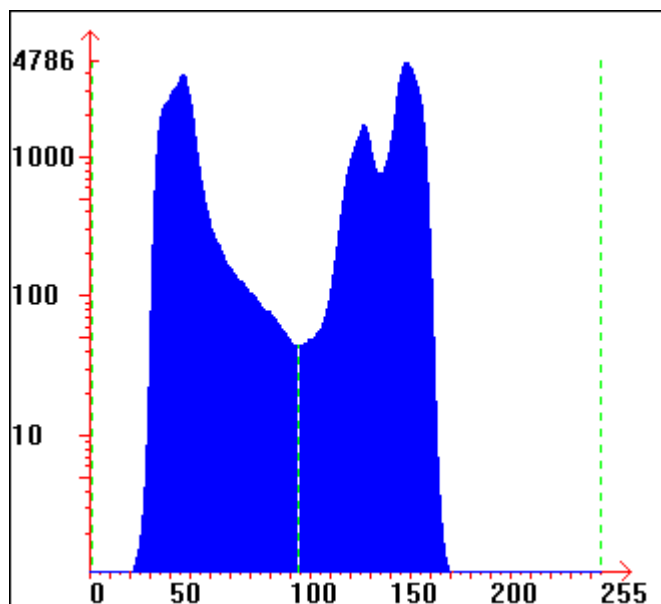


图 3.4 对数坐标直方图，两次平滑后，阈值 = 104

在实际情况下，图像常受到噪声等因素的影响，为了便于判定谷底位置，必须对直方图进行平滑。采用低通滤波算法对直方图进行两次平滑，这样可以保证峰值位置误差较小。图 3.4 为滤波后的结果。

图 3.5 是根据图 3.4 的阈值对图 3.1 进行阈值分割的结果。即：根据阈值将图像二值化，将物体和背景置为黑白两色。对图像扫描一遍，灰度大于阈值的点置为 255，即白色；小于等于阈值的点置为 0，即为黑色。

由于物体上有高光，所以二值化后，在黑色物体上会有小白点，如图 3.5 所示。为了使形心计算的结果准确，我们必须将这些小白点填充为黑色。结果见图 3.6。

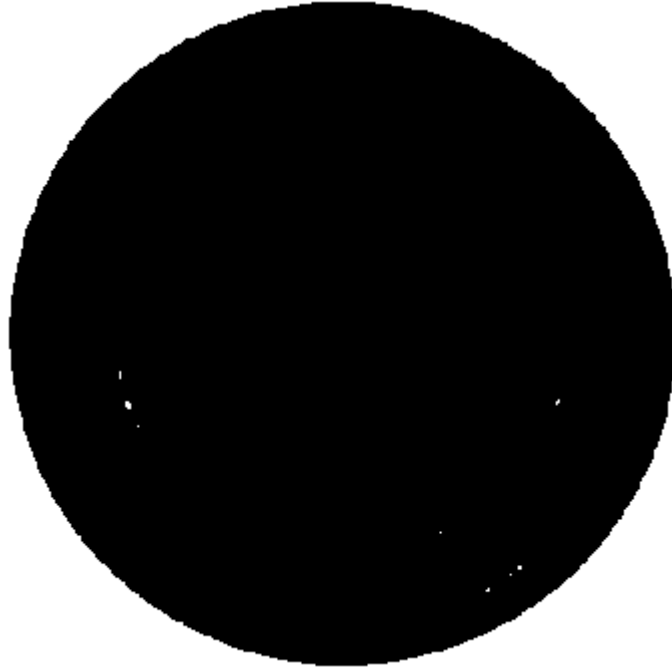


图 3.5 物体的二值化图像

为了能理解计算形心的公式，先看一个简单的力平衡的问题。如图 3.6 所示，一个质量忽略不计的刚性杆下有一支点，在支点两边分别在杆上放有 6 个质量均匀、尺寸相同、边长为 L 的正方体。设正方体的重量为 W ，当杆平衡时， $X = ?$

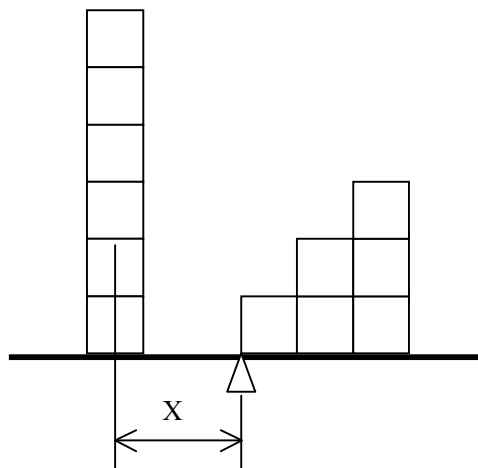


图 3.6 力矩平衡图

力矩平衡方程为：

$$6 \times W \times X = W \times 0.5 L + 2 \times W \times 1.5 L + 3 \times W \times 2.5 L$$

$$X = \frac{W \times 0.5 L + 2 \times W \times 1.5 L + 3 \times W \times 2.5 L}{6 \times W} = 1.83 L$$

X 为即为右边物体在水平方向的重心位置。

均匀物体的重心和形心重合。

计算平面图形的形心公式如下：

$$\bar{x} = \frac{\int x dA}{A}, \quad \bar{y} = \frac{\int y dA}{A}$$

将一个像素看成一个 dA ，且为 1 个单位，则图像的面积即为像素点的个数；分子上的积分则变为像素坐标的和。

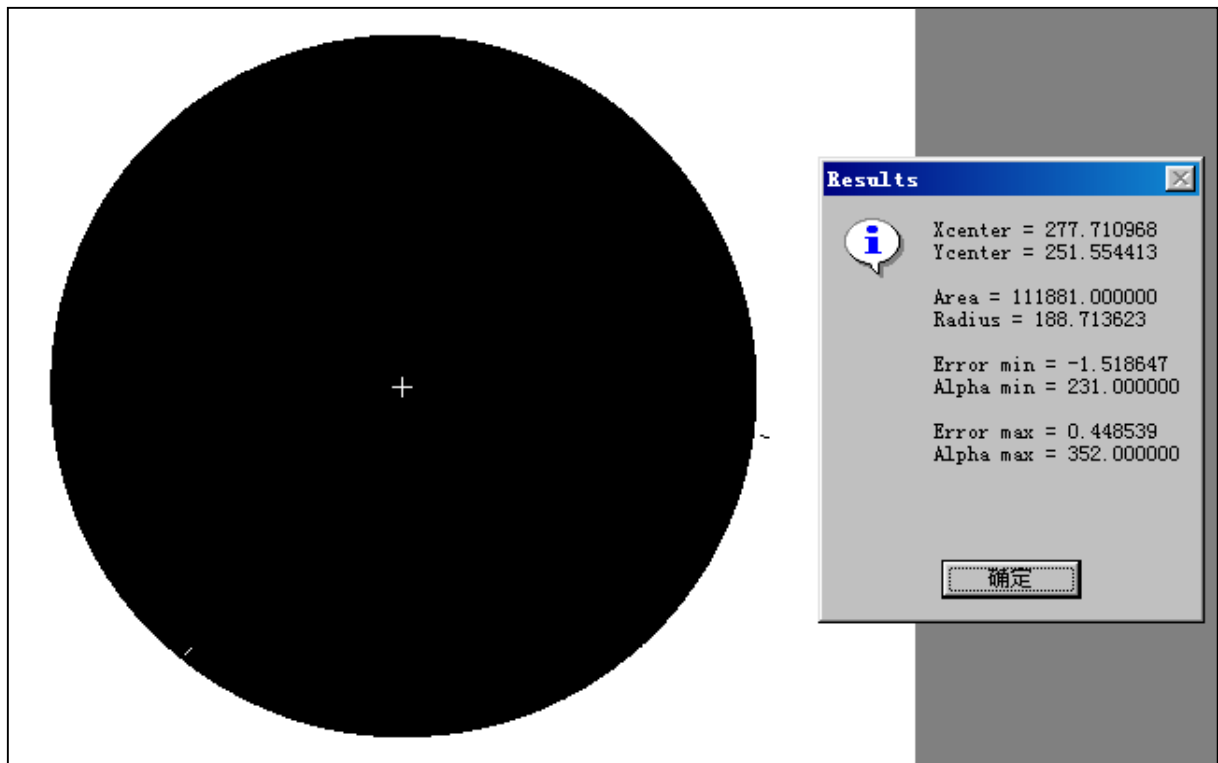


图 3.7 测量结果 1

图 3.7、图 3.8 给出了计算结果，形心位置用十字符号标出。

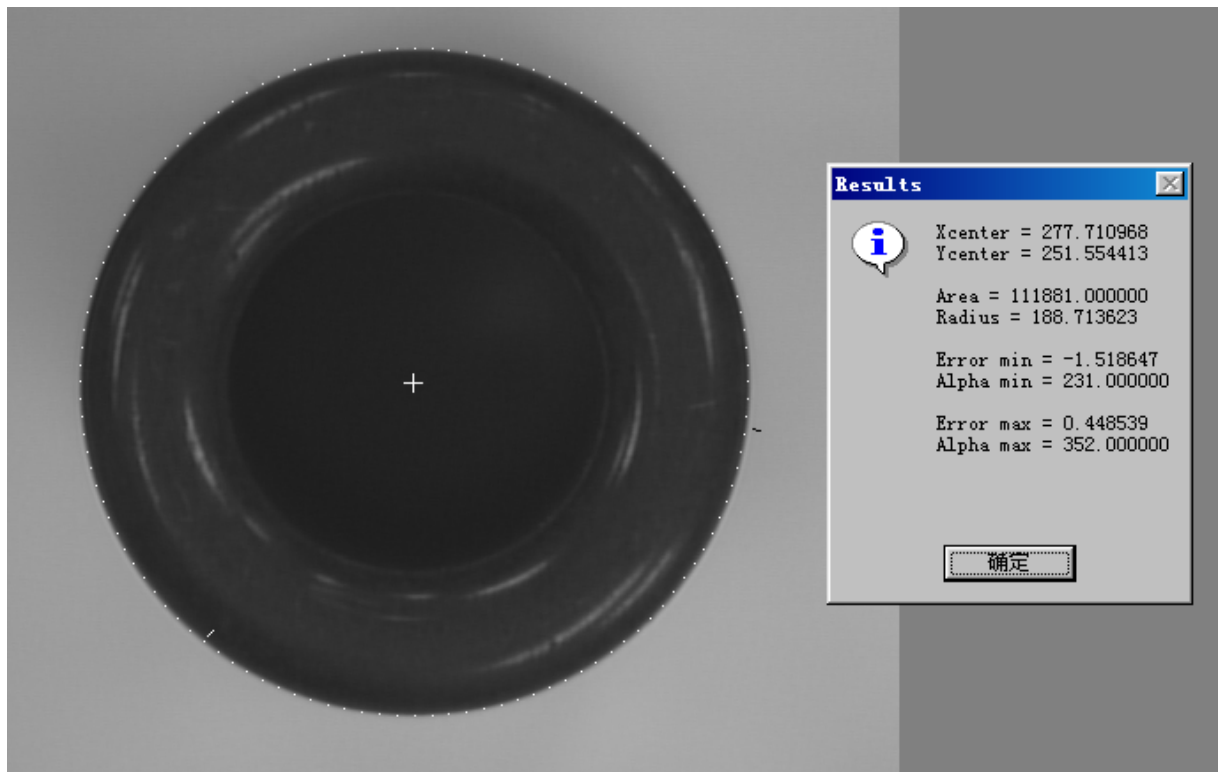


图 3.8 测量结果 2

得出面积再计算半径 R 就很容易了。因为 $R = \sqrt{A/p}$ 。

图 3.8 是在物体上用白点根据形心和半径画了一个圆，该圆和物体吻合，半径 $R = 188.7$ 个像素。

用图 3.7 的二值化图像上进行误差检查，最大正、负误差为 0.4 和 1.5 个像素，分别在 352、231 度处。

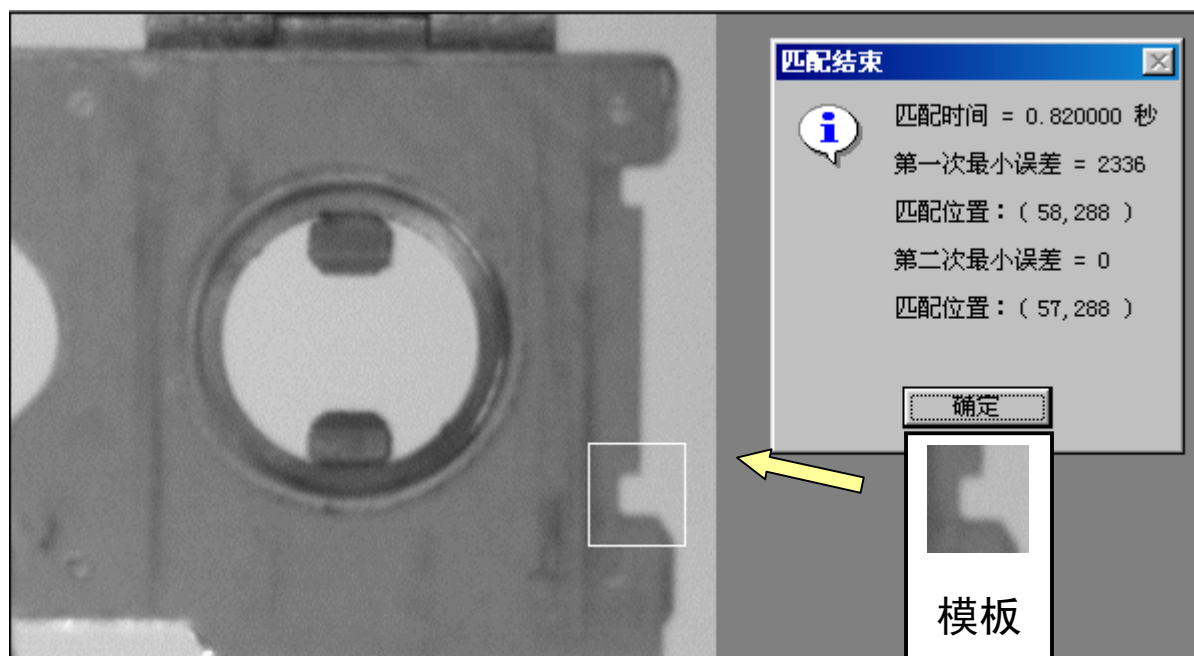
半径的最大误差 $< 0.8\%$ ，其中包括物体本身的不圆度和测量误差。这个结果说明用以上方法进行阈值分割、形心计算的效果不错。

四. 图像的模板匹配

1. 模板匹配的概念

模板就是一幅已知的小图像。

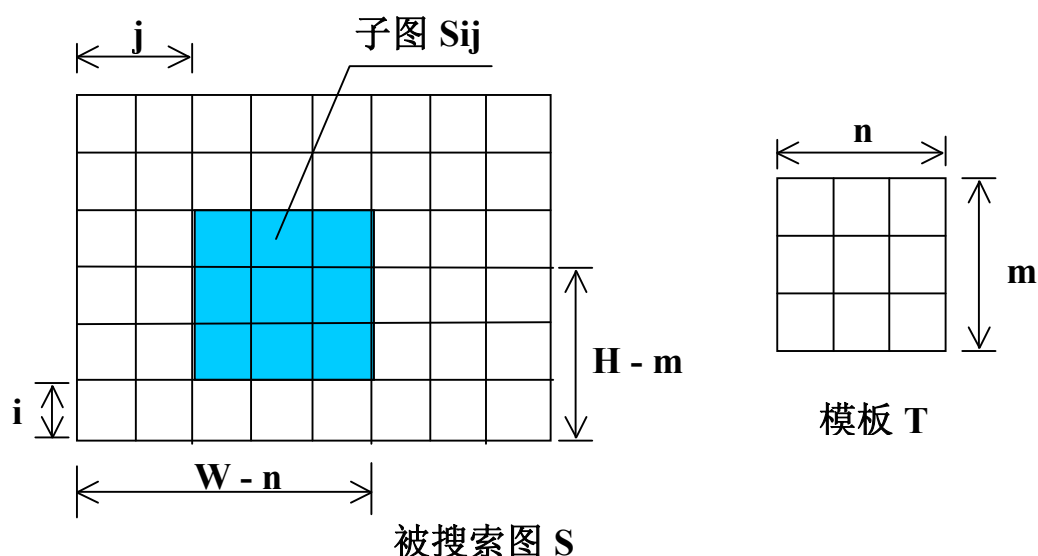
模板匹配就是在一幅大图像中搜寻目标，已知该图中有要找的目标，且该目标同模板有相同的尺寸、方向和图像，通过一定的算法可以在图中找到目标，确定其坐标位置。



以 256 级灰度图为例，模板 T ($m \times n$ 个像素) 叠放在被搜索图 S ($W \times H$ 个像素) 上平移，模板覆盖被搜索图的那块区域叫子图 S_{ij} 。 i, j 为子图左下角在被搜索图 S 上的坐标。搜索范围是：

$$\begin{cases} 1 \leq i \leq W - m \\ 1 \leq j \leq H - n \end{cases}$$

通过比较 T 和 S_{ij} 的相似性，完成模板匹配过程。



衡量 T 和 S_{ij} 相似性的公式

T 和 S_{ij} 的误差可用 D(i,j)描述:

$$D(i, j) = \sum_{m=1}^M \sum_{n=1}^N [S^{ij}(m, n) - T(m, n)]^2$$

$$= \sum_{m=1}^M \sum_{n=1}^N [S^{ij}(m, n)]^2 - 2 \sum_{m=1}^M \sum_{n=1}^N S^{ij}(m, n) \times T(m, n) + \sum_{m=1}^M \sum_{n=1}^N [T(m, n)]^2$$

上式的第一项为子图的能量，第三项为模板的能量，都与模板匹配无关。第二项是模板和子图的互相关，随(i, j)而改变。当模板和子图匹配时，该项有极大值。将其归一化，得模板匹配的相关系数，或称为匹配率：

$$R(i, j) = \frac{\sum_{m=1}^M \sum_{n=1}^N S^{ij}(m, n) \times T(m, n)}{\sqrt{\sum_{m=1}^M \sum_{n=1}^N [S^{ij}(m, n)]^2} \sqrt{\sum_{m=1}^M \sum_{n=1}^N [T(m, n)]^2}}$$

当模板和子图完全一样时，相关系数 R(i, j) = 1。

用相关系数 R(i, j)的公式可以做模板匹配，但运算量大。

另一种算法是衡量 T 和 S_{ij} 的误差，其公式为：

$$E(i, j) = \sum_{m=1}^M \sum_{n=1}^N |S_{ij}(m, n) - T(m, n)|$$

在整个图像中，E(i, j)为最小值处即为匹配目标。

2. 加速匹配速度的方法 ¾¾ 二次匹配

第一次：粗略匹配

a. 取模板的隔行隔列数据，即四分之一的模板数据，在被搜索图上进行隔行隔列扫描匹配，即在原图的四分之一范围内匹配。由于数据量大幅度减少，匹配速度显著提高。

b. 取一个误差阈值 E0，当 E(i, j) > E0 时，就说明误差太大，可以停止该点的计算，继续下一点计算。

为了合理的给出一个误差阈值 E0，我设计了一个确定误差阈值 E0 的准则：

$$E0 = e0 * (m+1) / 2 * (n+1) / 2$$

式中：e0 为各点平均的最大误差，一般取 40~50 即可；

m, n 为模板的长和宽。

第二次：精确匹配

在第一次误差最小点(i_{min}, j_{min})的邻域内，即在对角点为(i_{min}-1, j_{min}-1), (i_{min}+1, j_{min}+1)的矩形内，进行搜索匹配，得到最后结果。

下表是相关法、误差法、二次匹配误差法这三种模板匹配算法对两幅图像进行模板匹配的结果比较。

被搜索图 尺寸	模板名称	模板 大小	相关法 时间	误差法 时间	二次匹配 误差法时间
256*256	corner	16*16	0.94 秒	0.55 秒	0.06 秒
	eye	13*11	0.55 秒	0.38 秒	0.03 秒
	hell	15*22	1.16 秒	0.44 秒	0.05 秒
640*480	finger	13*17	4.12 秒	2.30 秒	0.22 秒
	nose	11*17	3.41 秒	2.20 秒	0.16 秒

五. 边缘检测、直线检测

5.1. 边缘检测

在对图像进行边缘检测前，先对图像进行阈值分割、二值化。结果见图 5.1 至图 5.3。

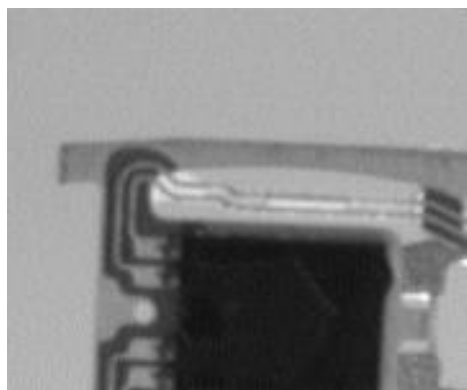


图 5.1 HGA 图像

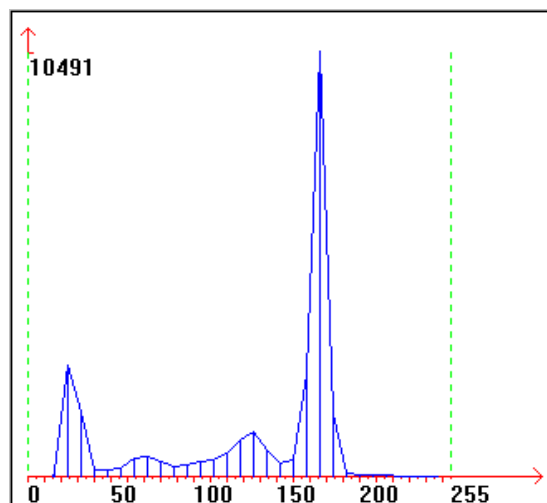


图 5.2 HGA 图像的灰度直方图，
阈值设为 155



图 5.3 二值化结果

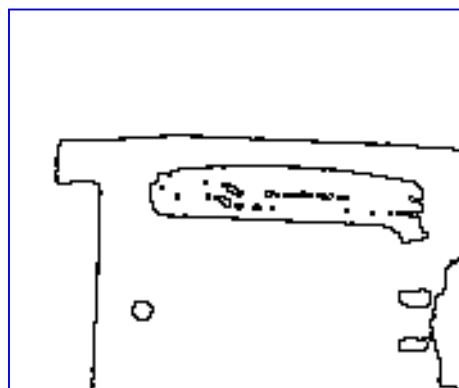


图 5.4 边缘检测结果

两个具有不同灰度值的相邻区域之间总存在边缘。边缘是灰度值不连续的结果，这种不连续可利用求导数方便地检测出来，如图 5.5 所示。

我们知道用梯度 $\text{grad } f$ 可以计算函数 f 各点的最大变化率，

$$\text{即 } \text{grad } f = \frac{\partial f}{\partial x} \mathbf{i} + \frac{\partial f}{\partial y} \mathbf{j}$$

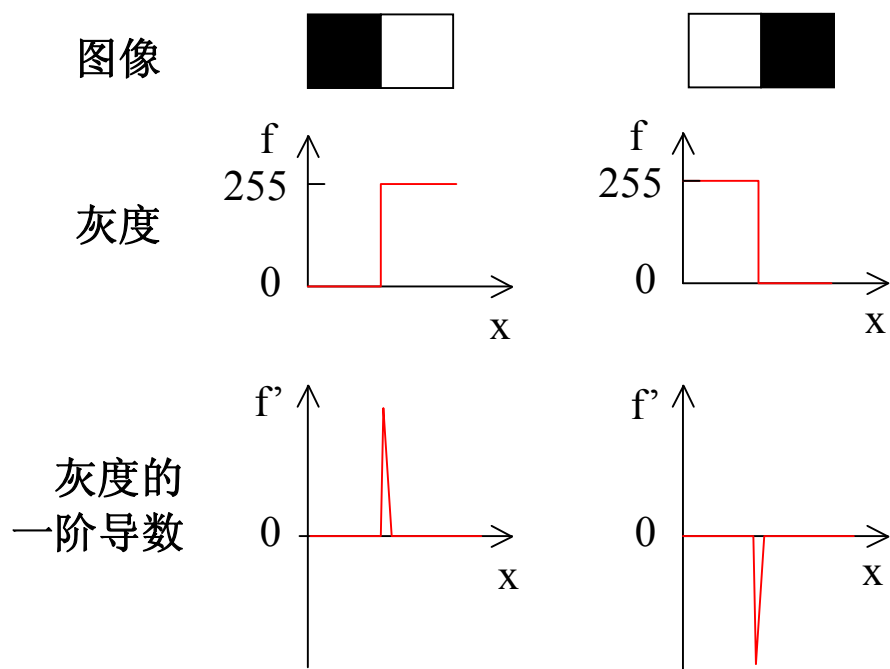


图 5.5 图像在 X 方向灰度的变化及导数

梯度的模为：
$$|grad f| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Roberts 边缘检测算法就是在上式的基础上变化而得：

$$g(x, y) = \sqrt{(f(x, y) - f(x+1, y+1))^2 + (f(x+1, y) - f(x, y+1))^2}$$

如果 $g(x, y) > 255$ ，则 $g(x, y) = 255$

$f(x, y+1)$	$f(x+1, y+1)$
$f(x, y)$	$f(x+1, y)$

图 5.6 Roberts 边缘检测算法中 4 个像素的位置关系

例：用 Roberts 边缘检测算法对表 5.1 中的数据进行处理，结果见表 5.2。

表 5.1 灰度图像数据

0	0	0	0	255	255	255
0	0	0	0	255	255	255
0	0	0	0	255	255	255
0	0	0	0	255	255	255
255	255	255	255	255	255	255
255	255	255	255	255	255	255
255	255	255	255	255	255	255

表 5.2 Roberts 边缘检测结果

-	-	-	-	-	-	-
0	0	0	255	0	0	-
0	0	0	255	0	0	-
0	0	0	255	0	0	-
255	255	255	255	0	0	-
0	0	0	0	0	0	-
0	0	0	0	0	0	-

图 5.4 就是用 Roberts 边缘检测算法对图 5.3 进行处理、并经过反色后的结果。

5.2. 直线检测

哈夫（Hough）变换可以检测已知形状的目标，而且受噪声和曲线间断的影响小。下面介绍采用 Hough 变换，在二值化图像中检测直线，确定其参数的方法。

哈夫变换的基本思想是利用点-线的对偶性。如图5.2.1所示：

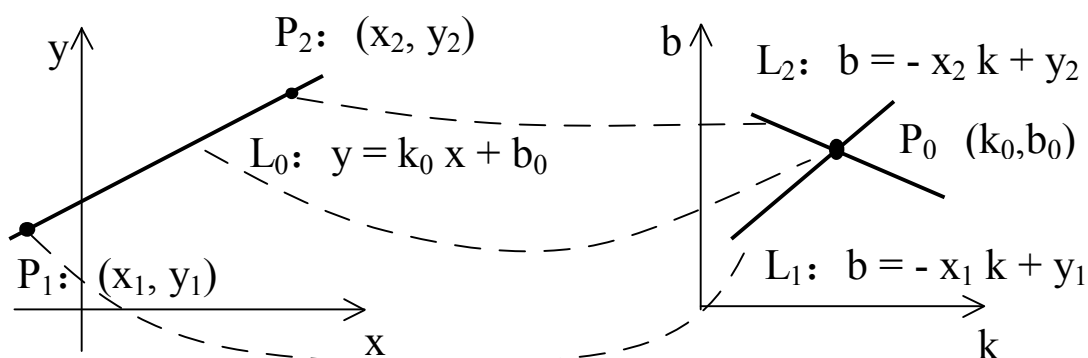


图 5.2.1 点-线的对偶性

由于 x-y 坐标中的垂直线的 k 值为无穷大，给计算带来不便，故使用点-正弦曲线对偶变换解决这一问题。直角坐标 X-Y 中的一点(x,y)，经过点-正弦曲线对偶变换：

$$p = x * \cos(a) + y * \sin(a)$$

在极坐标 a-p 中变为一条正弦曲线，a 取(0 ~ 180°)。可以证明，直角坐标 X-Y 中直线上的点经过 Hough 变换后，它们的正弦曲线在极坐标 a-p 有一个公共交点，如图 5.2.2 所示。

$$\text{当 } a \neq 0 \text{ 时, } k = -\cotg(a)$$

$$b = p / \sin(a) ;$$

当 $a = 0$ 时, $k = \infty$, 直线为垂直线。

$x = p$ (参见图 5.2.3)

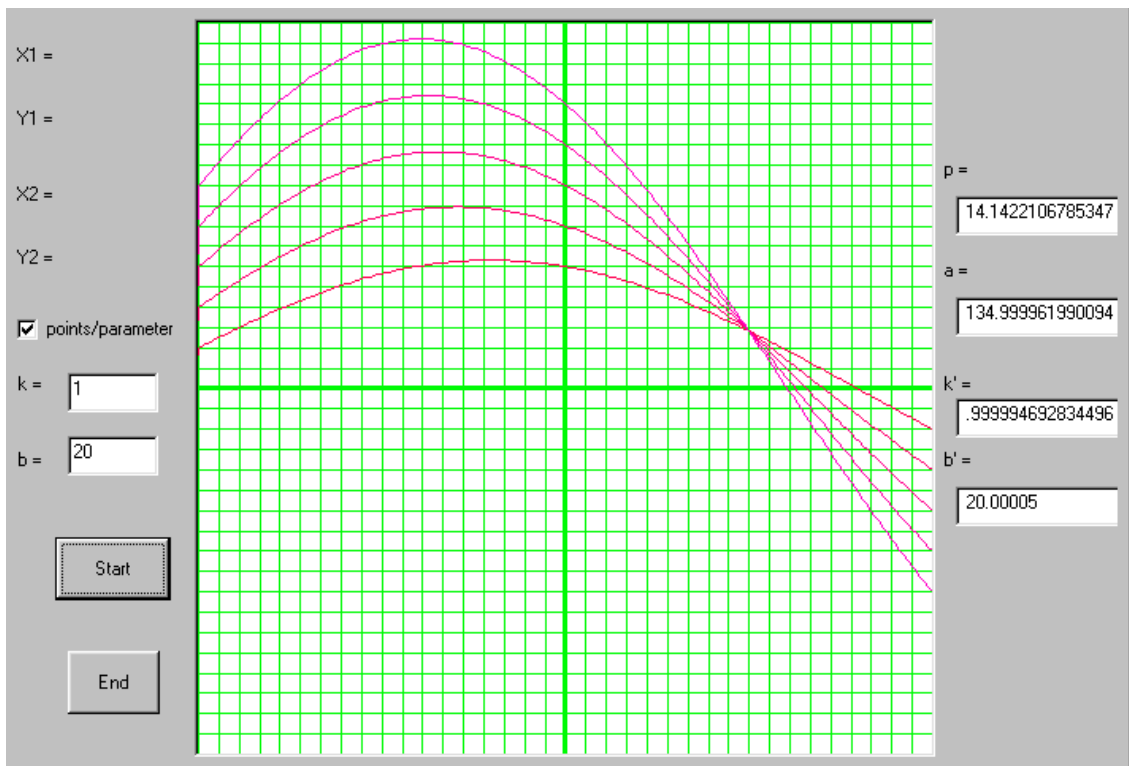


图 5.2.2 直线 $y = x+20$ 上的 5 个点
对应极坐标 $a-p$ 中的 5 条正弦曲线

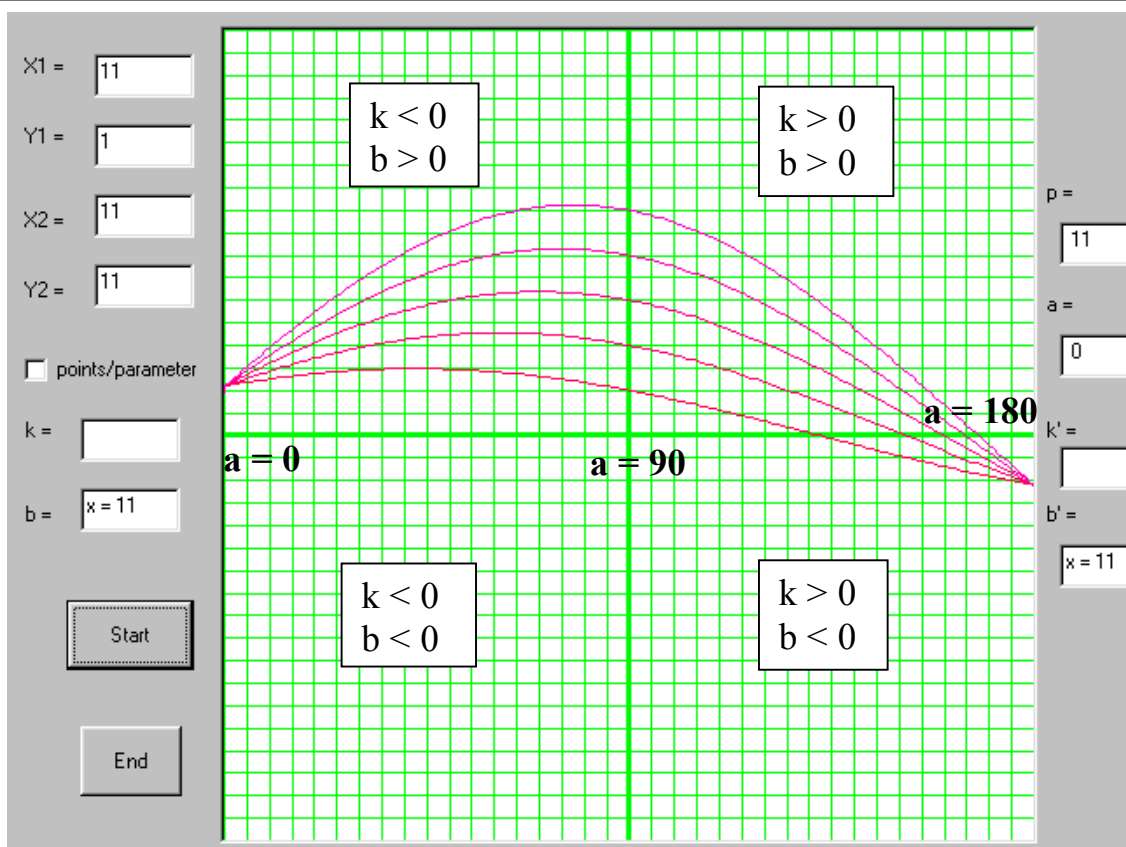


图 5.2.3 直线 $x = 11$ 上的 5 个点对应在极坐标 $a-p$ 中的 5 条正弦曲线，以及四个区域与 k 、 b 的取值关系

5.3. 两条直线的检测及其夹角算法：

- 在原图中找黑点；
- 根据黑点坐标在 ap 平面画正弦曲线，曲线经过之处，计数器加 1；
- 在 ap 平面里找最大值（参见表 5.3），根据其坐标计算第一条直线的 k_1 、 b_1 ；
- 将最大值的邻域清零，一般邻域取 $\pm 5 \sim \pm 10$ 个单位即可；
- 再次在 ap 平面里找最大值，根据其坐标计算第二条直线的 k_2 、 b_2 ；
- 计算两条直线的夹角 Φ ， $\text{tg}(\Phi) = (k_1 - k_2) / (1 + k_1 * k_2)$ 。若遇 90 度直线则不能套用此公式。

例 1 结果见图 5.3.1。

表 5.3 例 1 的 ap 平面里最大值的邻域数据

p \ a	86	87	88	89	90	91	92	93	94	95	96
38	18	14	9	4	4	5	114	83	44	36	38
39	17	13	6	5	8	7	96	50	36	13	20
40	10	12	4	7	5	9	137	54	18	6	12
41	7	9	8	6	11	39	120	63	6	13	12
42	7	8	5	7	9	162	72	10	3	26	11
43	10	5	8	10	38	192	16	2	3	29	17
44	8	8	6	9	97	151	11	3	3	18	17
45	9	6	11	29	91	74	34	2	3	10	13
46	13	11	13	38	82	32	7	3	9	13	13
47	7	12	40	72	74	15	2	4	29	11	14
48	14	18	22	66	74	14	3	2	28	17	17

在为了提高计算精度，又节约时间，可采取两次检测法。

- n 第一次全局检测，a 的分辨率为 1 度、p 的分辨率为一个像素，找到一直线，得其参数 k、b；
- n 第二次在此直线的邻域进行检测，a 的分辨率为 0.1 度。再次求其 k、b。
(第二次局部检测在第一直线的邻域进行检测，如图中灰色带所示。)

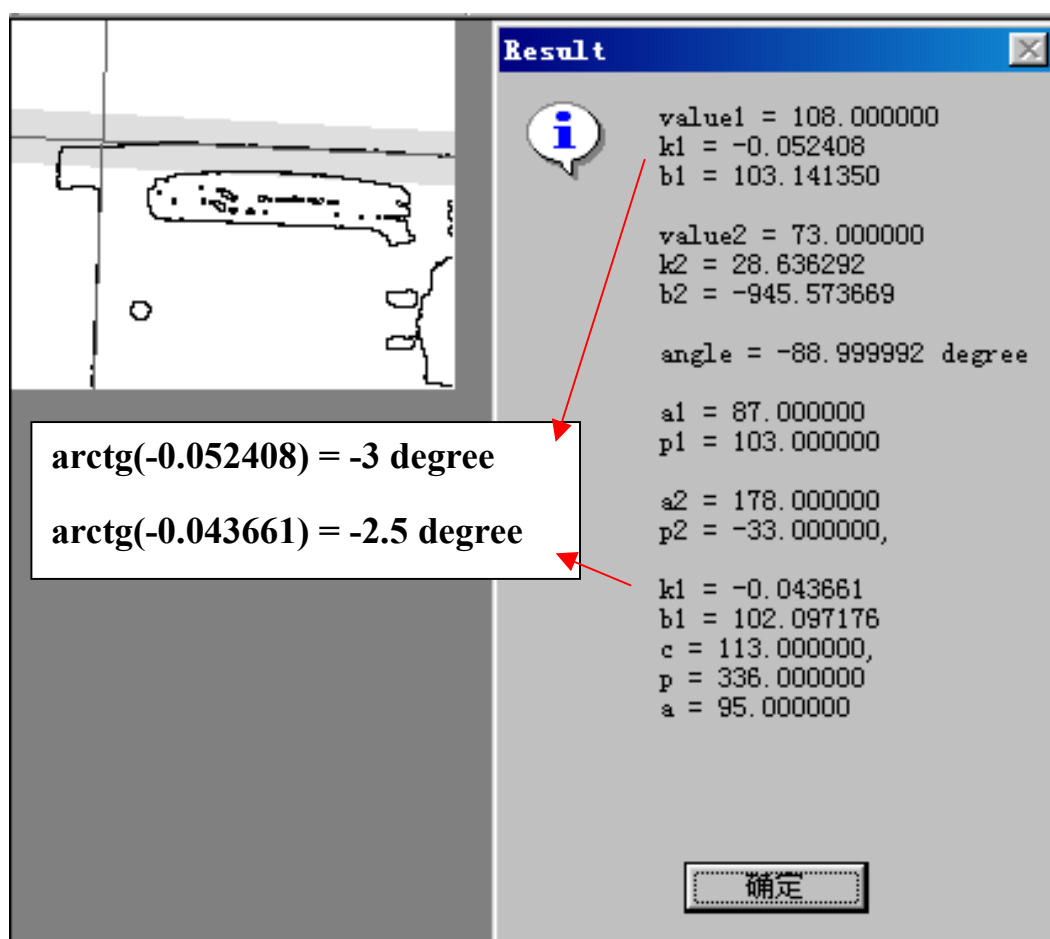


图 5.3.1 直线检测结果

注：角度的精度为 0.1 度，意味着：

573 个像素长的直线，当角度误差为 0.1 度时，其一端的误差为 1 个像素。

