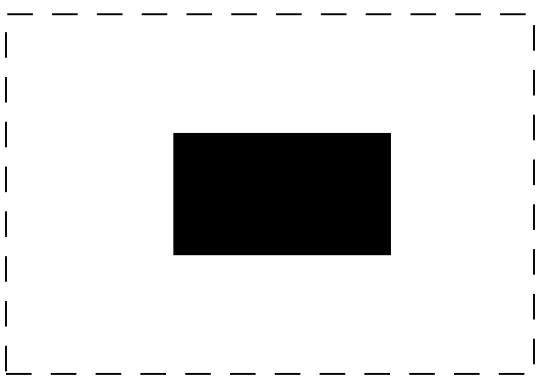
关于“边缘寻找”算法

TOM
本文转自 TOM 博客 http://www.blogcn.com/u/33/63/mvtom/index.html

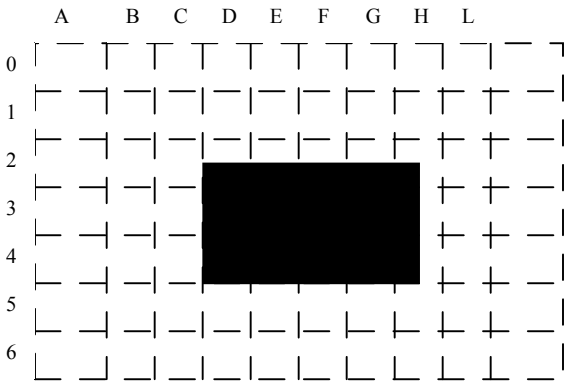
一、临界灰度值＋ 亚象素

现在假设我们要在图象中测量物体的长度。（如图＃1 中显示，虚线内为图象范围，图中背景为白色，被测物呈黑色）。



图＃1

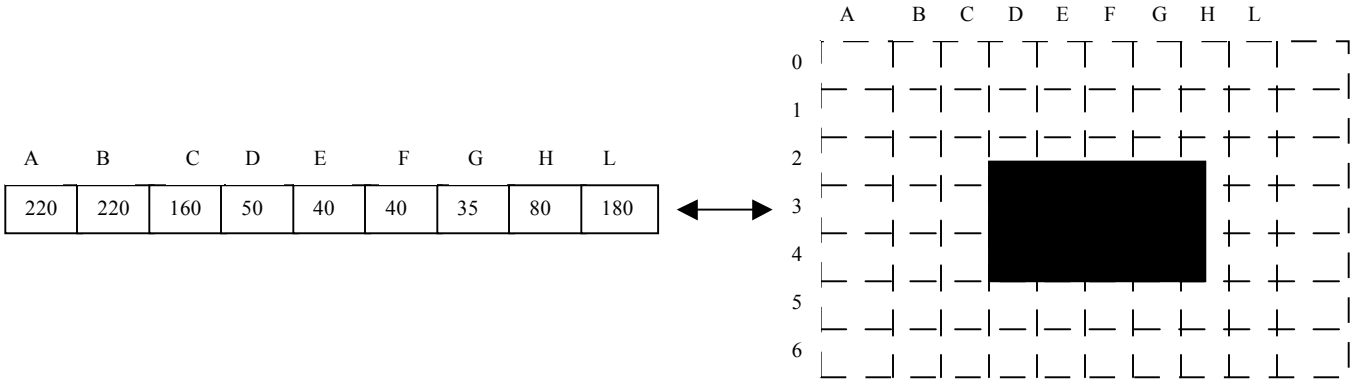
在相机拍照后，将图象视频信号传至视觉卡，由视觉卡把波状视频信号翻译成数字信号，存到电脑的内存中去（如图＃2 中显示，图象中的虚线格子为象素单元）。



图＃2

1，系统自我教育

接下来第一步，我们要先选取一个临界灰度值（threshold，有关这个名词的解释，请参阅主题文章“视觉 系统速成”）。大家已经知道，当图象转换成数字信号存到内存中去之后，我们便可以轻易地读取任何象素的灰度值。比如，我们现在从图象中拿到行 3 的全部象素灰度值（如图＃3 显示）。为了方便，这里我们假定此系统的灰度值的分辨率是 256 级，0 为最黑，255 为最白。



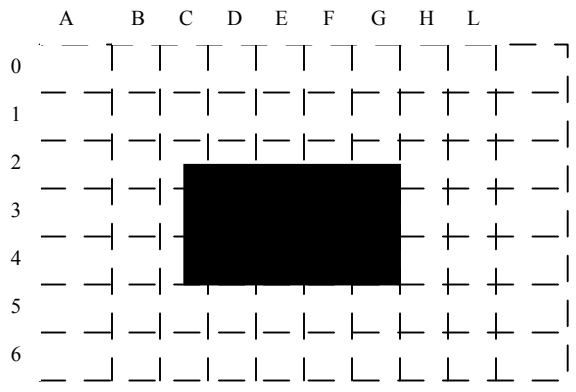
图＃3

从图中，我们可以看出被测体的范围（X 轴方向）是从列 D 至列 H。在行 3 的灰度值中，从列 D 到列 H 的灰度值分别为：50，40，40，35，80。由此可知：被测物体在成象中灰度值最高（即颜色最白）的象素的灰度值是象素（3，H）的灰度值，即 80。那么我们便可以认为，系统的临界灰度值就是 80，当然在实际测量中为留一定余地，我们不妨设为 100。（有关临界灰度值设定的具体方法，以后另题讨论。另：若成象中被测物为白色，则相反……选颜色最黑的地方的象素值为临界灰度值）。

设定临界灰度值后，系统便完成了测量的第一步：自我教育（通常这一过程英文叫 teach 或 training，此处是我乱翻的）。假如你已经完成了系统的标定工作（包括相机标定、系统标定），那么你就可以正式开始测量了。

2，测量

现在测量开始。我们把一个待测的被测物放到相机下拍照。在内存中所得图象如图＃4 所示。这里请大家注意，图＃4 与图＃2 略有不同。我只是为了让大家知道在实际操作中，每次拍照所得到的图象，必定是有所不同的。



图＃4

得到图象后，我们便可以让软件对图象进行分析。仍是以行 3 为例，假设系统是从行 3 开始进行逐点扫描的。扫描的过程如下：读取象素（3，A）的灰度值来与刚才所设定的临界灰度值作比较。如果象素（3，A）大过临界灰度值，也就是说象素（3，A）的颜色比临界灰度值白。那么这个象素就不是被测物上的一个点，于是就进行下一步的运算，读取象素（3，B）的灰度值重复上述运算，如此循环直至图象尽头，列 L。基本程序是

```
for (I=A; I<=L; I ++)
{
     if(pixel[3,I] <= 100)
    {
          flag = TRUE;
          break;
    }
}
```

程序中的 100 就是刚才系统所设定的临界灰度值。Flag 是一个标志，当系统找到被测物的第一个边缘时，flag = TRUE，于是系统便开始寻找第二个边缘。在本例中，第一个边缘是由白到黑，第二个边缘是由黑到白。

假设在新的图象中，行 3 的灰度值如图＃5 如示。那么系统扫描到象素（3，D）时，程序便会中断，flag = TRUE， 并开始寻找第二个由黑到白的边缘。第二次寻找基本程序如下：

A	B	C	D	E	F	G	H	L
220	220	120	50	40	40	60	180	180

图＃5

```
for (J=I; J<=L; J ++)
{
     if(pixel[3,J] >= 100)
    {
          flag = FALSE;
          break;
    }
}
```

按图＃5 所示，第二边缘的寻找，进行到象素（3，H）时，便自动停止。现在我们根据程度运行时中第一个变量 I 所记录的数值，知道第一个边缘为 D。由变量 J 知道第二个边缘是 H－1＝G。由此，可以知道整个被测物 X 轴方向的大小＝G－D＝4 个象素。4×象素值（由系统标定得到），我们就最终得到被测物体的左右长度了。

但事情并没有这么简单，现在真正的问题出现了。如果大家细心的话，便可以发现在图＃4 中，被测物实际上从列 C 开始的。只不过在列 C 中，被测物只有一部分，而另一部分是白色的背景。下面，我们就要来计算一下这个“一半一半”的部分，也就是大家期待已久的“亚象素算法”。

2，亚象素算法

亚象素的基本算法其实并不难，不知道为什么大家问个不停？难道国内没有相关资料介绍？不明白！

一言以蔽之，亚象素基本思路就是将一个象素再分为更小的单位。在我们上面的讨论中，一直以 8bit 的系统作例子，也就是说一象素的灰度值分为 256 级。所以，以这类系统为例，作亚象素计算就要把象素分为 255 个小单位。

或许，可以这样来理解“亚象素算法”。一个象素的灰度值从 0 到 255，0 是纯黑，255 是纯白。我们来把象素想象成是一个由 255 个小象素所组成的集合。而每个小象素都是一个独立的小镜子，那就是说一个象素里面有着 255 个小镜子。灰度值则可以想象是表示有多少个小镜子反光。0，的意思就是 255 个小镜子全都没有反光；255 的意思就是，255 个镜子一起反光。上面讲到的所设定的是临界灰度值为 100，就是说 255 个镜子中有 100 个在反光，另外 155 个镜子没有反光。

现在，回到上面的测量例子中来。从图＃4 中可以看到，被测物在列 C 中只有一半，而正是由于这个原因所以列 C 的灰度值是高于临界灰度值的。只要我们把这部分也算到最终测量数据中去，所得到的结果必定更为准确。由此大家可以知道，真正的计算被测物的长度公式，并不是（象素数量×象素值）这么简单。而应该是（（象素数量＋第一边缘亚象素值＋第二边缘来象素值）×象素值），具体到本例，被测物左右真实长度＝（（4＋象素（3，C）亚象素值＋象素（3，H）亚象素值）×象素值）。

说了半天到底怎么算亚象素值呢？非常简单，亚象素值（白色部分）＝该象素灰度值/256；亚象素值（黑色部分）＝1－亚象素值（白色部分）；仍以图＃4 为例，象素（3，C）的亚象素值＝1－（120 / 256）＝0.53； 象素（3，H）的亚象素值＝1－（180/256）=0.3。而整个被测物左右实际长度为 4.83 个象素。其实就是在算有几个镜子在反光，有几个没反光罢了。

好了以上就是亚象素的基本算法。在结束这个算法讨论之前，有两点我必须向大家强调一下：一、在实际情况下，大家不可能看到图＃4 中所显示的情况－－象素的一半是黑色另一半是白色。这只是为了方便大家理解所画出来的。而真实的情况是一个象素就只是一小块灰色，没有明暗的分别。明暗的区别只能在象素与象素间显现出来；二、在描述亚象素的基本算法时，我所说“小镜子”的概念完全是本人随口胡说，大家绝不会在任何有关图象处理的理论著作上看来这类胡言乱语。这些同样是为了方便大家理解而讲，当然也因为本人水平不限，不能以纯数学语言来表达。

2，亚象素算法的几种变种

第一种：亚象素值（白色部分）＝（该象素灰度值×（临界灰度值/256））/256
　　亚象素值（黑色部分）＝1－亚象素值（白色部分）
第二种：亚象素值（白色部分）＝后象素值/（前象素值 +后象素值）
　　亚象素值（黑色部分）＝1－亚象素值（白色部分）
第三种：亚象素值（白色部分）＝（象素值－前象素值）/（后象素值－前象素值）
　　亚象素值（黑色部分）＝1－亚象素值（白色部分）

至于这几个变种的亚象素算法，这里就不多解释了。如果哪位朋友有兴趣的话，我们可以另开辟新主题讨论。