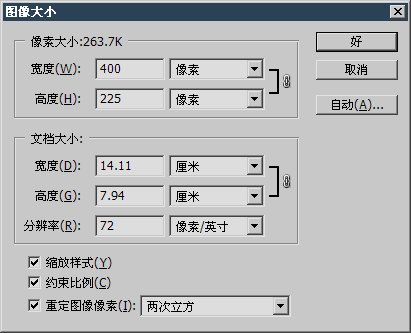
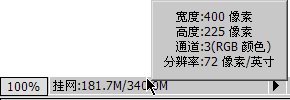
**第2章　第2小节　点阵格式图像**

我们所看到的图像，究竟是如何构成的呢？这就需要涉及到图像类型的概念。 电脑中的图像类型分为两大类，一类称为点阵图，一类称为矢量图。

点阵图顾名思义就是由点构成的，如同用马赛克去拼贴图案一样，每个马赛克就是一个点，若干个点以矩阵排列成图案。 数码相机拍摄的照片、扫描仪扫描的稿件以及绝大多数的图片都属于点阵图，如下左图就是一个典型的点阵图。把这幅图片调入Photoshop，使用菜单【图像 图像大小】就可以看到如下 右图的信息。注意窗口上部像素大小的宽度和高度，分别是400像素和225像素。 像素是什么？像素就是组成点阵图像中的那些点，是点阵图最小的单位。如同拼成图案的那许多马赛克一样。



在图像窗口底端状态栏的缩放倍数右方区域，按住ALT键单击就会出现像素数量信息，如下 左图。PhotoshopCS及更早版本的图像状态栏位于Photoshop窗口的底部，如下右图。如果状态栏未出现可通过菜单【窗口 状态栏】打开。PhotoshopCS2版本则没有此项菜单。



如果我们放大图像〖CTRL +〗，就会看到点也同时被放大了，这时就会出现所谓的马赛克现象(也称锯齿现象)。如下左图。我们可以看到有许多不同颜色的小正方形，那就是被放大的像素。每个像素只能有一个颜色。 宽400像素，高225像素，意味这幅图像由横方向400个点、竖方向225个点组成，400x225=90000，图像的总像素数量就是9万。 数码相机有一个很重要的指标就是1300万像素、3000万像素这样的称呼，就是指拍摄出来图像中的像素总量。

像素是不是越多越好呢？从大部分情况来说是的，图像的像素越多，记录的信息也越详细，图像的局部就越细致，如下右图。 上图是一幅像素总量144万(1600x900)的图像，在放大到和前一张图差不多的大小时的效果，可以很明显感觉出图像局部要显得细腻得多。



放大缩小图像的快捷键是〖CTRL +〗〖CTRL -〗，这种放大会沿着图像的中心点放大。

还有一种定点放大方式是按住空格和CTRL键，用鼠标单击图像的一个部分，这样会以单击的地方作为中心放大。缩小是按住空格和ALT键单击。 如果缩放程度不够可重复以上操作。图像窗口的标题栏以及状态栏都会显示缩放倍数。 严格来说应该是先按CTRL再按空格。但是在中文Windows下这正好是默认的中文输入法切换方式。因此建议先按空格再按CTRL。

如果图像超过了图像窗口的大小，将在右方和下方出现滚动条，此时拉动滚动条即可移动观看区域(不是移动图像)。 也可按住空格键在图像中按下鼠标拖动。当鼠标开始拖动以后，空格键可以松开。

以上的几个快捷键是最常用的。要把它们记住，这将让你的操作变得迅速。

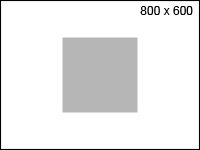
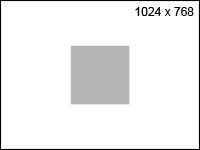
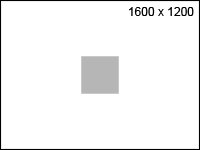
**0201**我们的显示器也是点阵式的，前一课中看过的电视机屏幕，就是由许多的点构成的。包括液晶屏和等离子屏也是如此。 传统的显像管显示器又称为CRT(学名阴极射线管)，是显示设备中最早也最普及的种类。 显示器的点阵数是可变的，我们可以从Windows控制调板中的显示项目来查看或改动目前显示器的屏幕分辨率，如下图所示，目前为1024x768像素，也就是说现在显示器横方向能够显示1024个 像素点，竖方向768个像素点。



如同一张6寸的照片不能完整放入一个5寸的像框一样，如果一幅图像超过了显示器横或竖方向的像素数，那么这幅图像就不能在屏幕上完整显示(以100%原尺寸显示前提下)。 因此屏幕分辨率越高，能够完整显示的内容就越多。比如一个300x300像素的方块，在不同的屏幕分辨率下，所显示的大小也不一样，如下3图。

如同马路上的一个井盖，站在5层楼看得很清楚，站在30层楼就小了许多，站在70层楼就更小了。井盖的实际大小没有变化，是视野放大导致井盖看起来缩小。

因此这不能说方块的大小改变了，300像素还是300像素没有变，是因为屏幕像素总量的增加使得它看起来变小。 计算一下就会知道，屏幕横分辨率800的时候，300像素占据着约一半的宽度，在1024时候占据约三分之一，在1600时候只占据约五分之一了。

显示器上有着许多的点，每个点由红绿蓝三个涂有荧光粉的单元组成，学名就是像素。 显示器内部有3个称为电子枪的部件是用来激发像素发光的，它发出3道电子束击打屏幕最左上角的点(可称为0点)，3束分别击打红绿蓝单元。 根据信号的强弱使红绿蓝按照比例混合成色。这样便完成了一个像素的激发。 然后在偏转线圈的磁力作用下，电子束往右移动一像素，再对这个像素进行激发。直到顺序激发完这一行所有的像素。就称为完成一个“行扫描”。 然后向下移动一行，回到左端开始激发第二行的第一个像素。如此重复进行。 当激发完最后一行最后一个像素后，就算完成了一个“场扫描”。

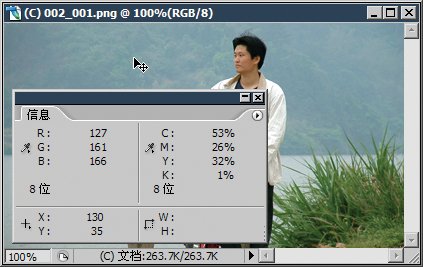
因此其实屏幕上的像素并不是同时发光的，但由于人眼有视觉暂留的特性，所以还是可以看到一幅完整的图像。 显示器扫描的顺序如下图，从0点开始往X轴方向扫描，完成一行后往Y轴方向移动，进行第二行的扫描。 正因为这种扫描方式，屏幕的坐标系就如同上图一样，以左上角为原点，X轴向右，Y轴向下。这与平面几何坐标系正好上下颠倒。



当我们把图像储存为点阵格式的时候，Photoshop就按照上面的顺序，一个接一个地去记录这幅图像中所有的像素的颜色。从而储存了图像。 当在Photoshop中打开一幅点阵图像的时候，也是按照上面的顺序逐个提取像素信息并显示在屏幕上。

我们浏览网页的时候，有时会看到图片不是一次性出现，而是从上至下慢慢显示出来。这就是因为网络传送速度慢造成了点阵像素信息的读取也慢。 如果等待传送的时间太长，会让浏览者失去兴趣。因此用于网页上的图片，应该尽可能地减少大小，便于快速地传送。这个要点我们在以后还会接触到。

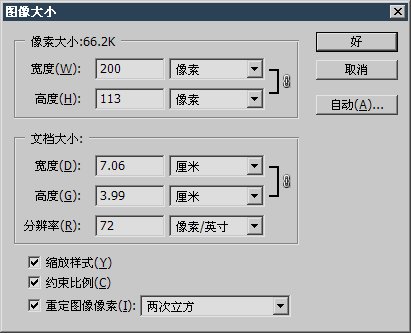
Photoshop的信息调板会显示当前鼠标在图像中的XY坐标，如下图鼠标的热点正处在这幅图像横方向第130个，竖方向第35个像素处。 所谓鼠标热点是指鼠标光标中起定位作用的那一点，不同的光标热点位置也不一样。下图的热点是在箭头顶部的尖角处。



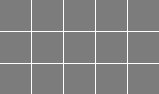
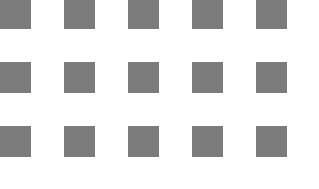
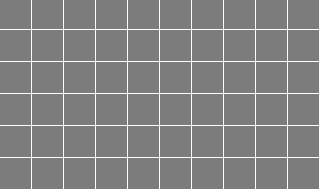
了解了点阵图像大小的区别后，我们知道点阵图像幅面越大，像素越多，记录的信息就越丰富。 但是如果只有小幅的图片，能不能放大呢？放大的效果以后比起真正的大图来如何呢？下面我们来做一个实验。

**0202**使用菜单【图像 图像大小】，将宽度改为200，注意高度也随着发生变化，这是因为下方的“约束比例”选项有效。这样可保持图片宽高比例不变。 对话框中其他的选项暂时不用去深究。参照下左图即可。

按下好按钮确定操作，会看到图像已经变小了。如下中图。 做完这一步后，我们将一个大图缩成了小图，从9万像素(400x255)变到了2.26万像素(200x113)。 现在再次使用菜单【图像 图像大小】，将图像宽度改回400。注意自动计算出来的高度是226而不是原先的225了。如下右图。



发现图像变得模糊不清了，原先可以看到的一些细节(如左手的手指缝)丢失了。这是为什么呢？ 首先我们来模拟一下第一次缩小的过程，假设我们要将一幅10x6个像素组成的图像，缩小为5x3，以下是示意图，每个灰色方块代表1像素。如下左图。当缩小指令发出后，Photoshop等距离地抽取像素并丢弃，如下中图。 然后再将剩余的像素拼合起来，形成缩小后的图案，如下右图。



在第一次缩小以后，像素从9万降到了2.26万，这其中丢弃了6.74万个像素信息。然后又将图像扩大到400×225像素，虽然像素总量和原先一样是9万， 但在第一次转换中丢弃的6.74万像素信息却是找不回来的。Photoshop只能采用插值算法去弥补这6.74万像素。 所谓插值算法，就好比猜测，凭空去“捏造”那些并不存在的像素。

如下图是上左图左上角那2x2的部分。 现有A、B、C、D四个像素，要将2x2扩成3x3，那么就要多出5个像素。图中的标号是1、2、3、4、5。 如何确定这原先并不存在的像素的颜色呢？是将现有两个像素的颜色值取平均，去作为新像素的颜色。 也就是说AB运算后得出1；AC运算后得出2；BD得出4；CD得出5；3则是由1245运算得出的。

注意：以上内容是为了便于大家理解而打的比方，真正的图像运算概念和过程远比这复杂的多。



可以想象，用这样方式“捏造”出来的像素，和真正原先的像素肯定存在误差甚至是很大的误差。 比如左手的手指缝，原先可以很清楚地看到有三道深色的线，如下左图。

而在缩小以后，原先手指缝部分的像素就几乎不见了。只剩下一些浅色的像素，手指缝已经看不清楚了，如下中图。

用这些浅色的像素计算出来的新像素，同样也只可能是浅色而不可能是深色的。因此手指缝原先深色的部分已无法还原了。如下右图。



失去了手指缝的分界，整个手看起来就变得模糊不清，图像失真了。这就是为何将小幅点阵图像扩大后，图像会变得模糊的原因。 在今后实际的操作中，一般情况下不要将点阵图放大制作。

现在回顾一下刚才第二次改变图像宽度到400的时候，高度是226。而不是先前的225。 既然是先缩小一半，再扩大一倍，等于是先除去2，再乘以2，数字应该和原先相等。为什么不相等了呢？

这是因为一个前面我们已经提到过，但是没有加以强调的概念：像素是最小的单位。它是不可再被分割的。

图像中的像素总数一定是一个整数，不存在500.7或者400.3这样带小数的数量。 那么，225÷2=112.5，Photoshop近似算作113像素了。而后第二次的扩大，是以这113作为基数，因此得出226像素。

我们前面所做的事情，用语言来表述是：将图像缩小一半，再扩大还原为原大小。对吗？这句话是错误的，有概念上的根本错误。

首先，扩大和还原是两码事，扩大是一种对图像进行修改的操作。而在Photoshop中，还原是指对上一步操作的撤销，对图像本身并没有再作操作。 好比你拒绝接收你朋友寄来的一封信，那么它将被原样不动地退回，那是对寄信这个操作的撤销。可以称为退信。 但如果你拆开信封阅读了内容之后，又把它装进新的信封，并且写上朋友的地址寄回，那么这已经是新的寄信操作，而不能称为是退信了。

再者，“原大小”中的“原”也是错误的，通过上面的分析，我们知道由于取整数的问题，图像的大小和原先已经不一样了。

有关Photoshop的撤销操作，将在后面的内容中介绍。

现在再概述一下点阵图像：

点阵格式是把图像分为若干个点(像素)，依靠储存或再现每个点的信息，从而储存或再现整幅图像。

由于像素数量的限制，所以点阵图像的大小是固定的。缩小或放大图像都会造成对图像的破坏。