

如何将爱因斯坦画到墙上

此文件是科普性的文章，最初由本店发表在知乎。文章介绍了墙画机的原理，可以帮助读者迅速了解墙画机工作原理。但并非使用说明，具体的拼装、程序调试，请阅读《Wall Drawing Machine 安装调试说明》文档。

疑问，为什么要画爱因斯坦？其实画人，画动物，画山水都一样，我只是找了一个抓眼球的名人而已，不要较真，画谁都一样。

提到墙画，你是不是想到了这些



如果这个太土了



那我们换个雅一点的？



或者是细致一点的？

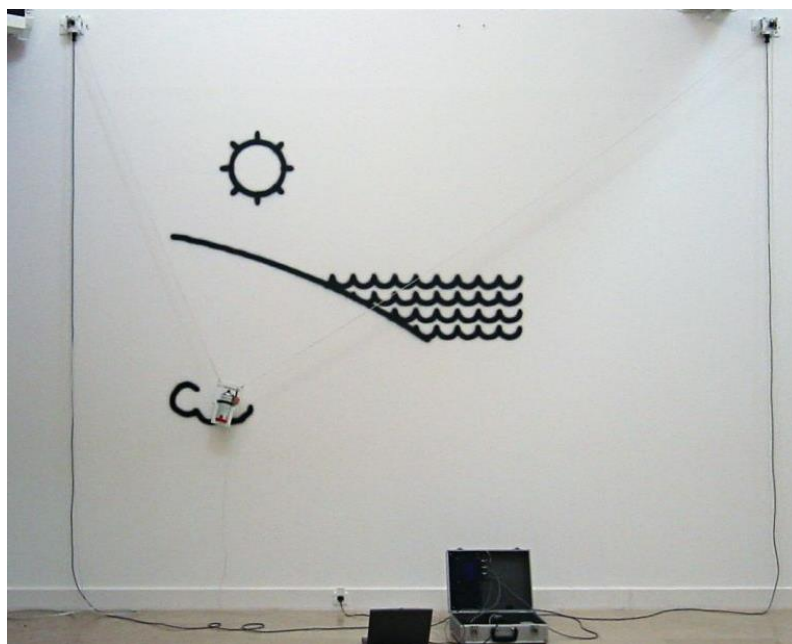
管他什么画风呢，这些不重要，不管他们画风如何，都是练过的。关于绘画基础我们大部分人都一样，美术课都被物理老师霸占了，还能说啥呢。完全靠手工是不行不通的。



既然手工不行，那就上机器！墙体打印机了解一下？

只需 38888¥ 整机送到家，上午培训，下午出图，当天赚钱，2 个月回本，1 年奔小康。好像除了贵点，也没什么槽点。要不我们自己动手做一个墙画机怎么样？

思路是这样的，找个喜欢的画，用机器把它画到墙上。完工！显然不能这么潦草，说细一点吧。既然不想搞那么多铝型材，搭架子，还要上各种步进电机、驱动器、喷墨头、墨水。。。我们搞简单点。弄一支笔直接画在墙上。至于笔怎么控制么。拿 2 条绳子栓住了，拉绳是不是也可以？



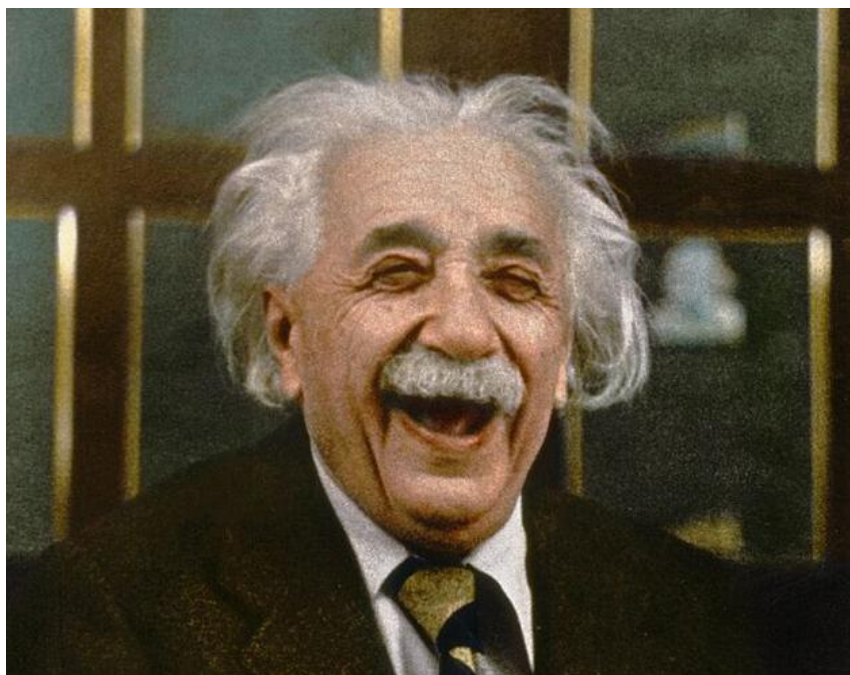
早在 2002 年，有两位天才设计师搞出了一个叫 Hektor 的设备。

它由 2 个电机控制 2 条绳子拉着一个自喷漆，在墙上喷漆作画。后来这个结构的设备，在创客圈广泛传播，有很多名字 V Plotter Design，Polargraph，Wall Drawing Machine，Drawbot 什么的，反正叫啥不重要，2 个电机 2 条绳（也有 4 电机，下方还有 2 个，这种结构可以放平在地面上用），拉着一个可以画线的设备移动就是了。绳子牵引支架在墙上移动，可以拉扯到墙体的大部分区域（实际上很多位置是无法达到的，最佳的绘图区域是个相对较小的范围。）再有一个可以控制落笔，抬笔的机构，即使不能抬笔仅能一笔画完，也可以考虑狂草书画模式吗，哈哈。以上就是它的硬件工作原理部分了。

我们需要一个可以控制电机正反转的 Arduino 或者树莓派，51，STM32，PLC 不管是啥，能控制电机转的东西就行。除了 Arduino 另外几个我都不会用，所以。。。。。。其实他们的原理都差不多，程序上也没有太大区别的。

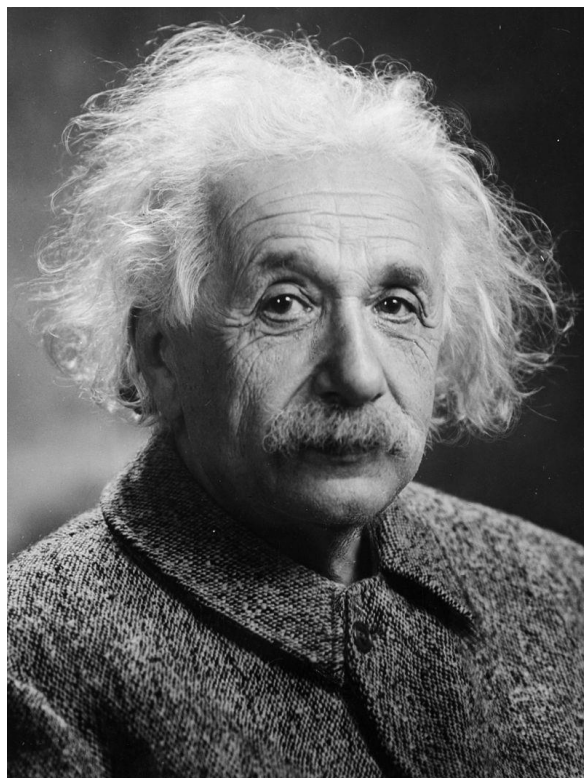
既然原型机有了，工作原理也了解了，想想成功的众泰汽车。那我们还等什么？操练起来吧骚年，莫等闲。让我们开始画爱因斯坦吧。

第一步：Google 一张爱因斯坦的照片下载下来。



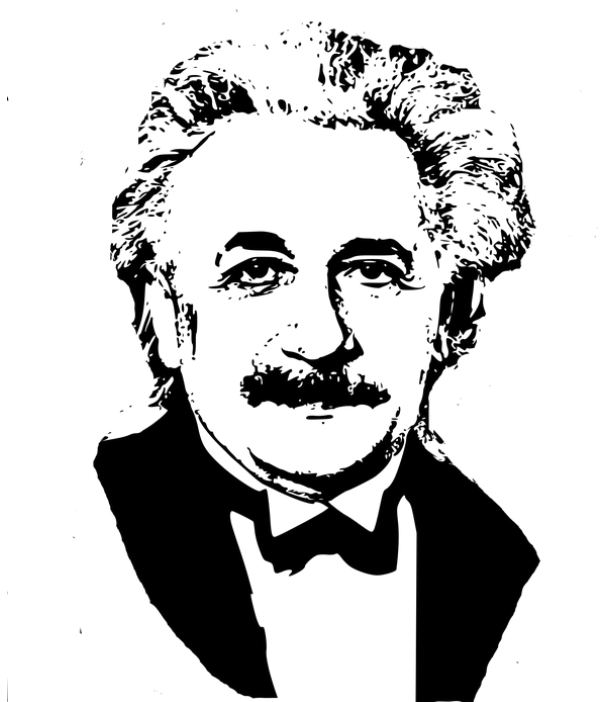
遇到了障碍，真是完事开头难，我们只有一直笔，虽然可以是彩色笔，但是依然只能绘制一种颜色，所以我们需要找一张单色的照片才行。其实我们可以开发一个 12 色笔支架，选用不同颜色的笔作画，甚至 256 色的笔架。。先不做梦了，搞单色的先。多色是四期工程（二期，三期工程是啥？？还没想好）

第二步：重新 Google 一张单色的爱因斯坦

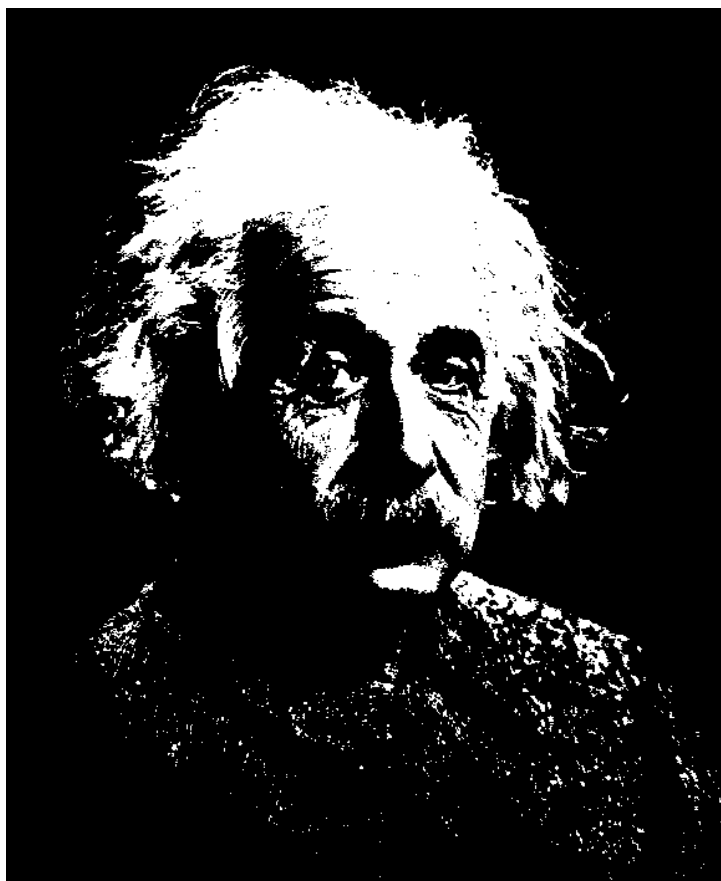


等等，又遇到了困难，虽然照片是黑白的了，但是照片里依然有很多个不同的灰色，深灰，浅灰，深深灰，浅浅灰。。。我们的笔，显然还做不到不同深度的笔触。（其实，我们可以考虑给笔施加不同的压力，让画作有笔触的感觉，压感屏就可以根据力度画深浅的。二期工程有了）所以我们需要一张纯黑白的不要灰的照片。

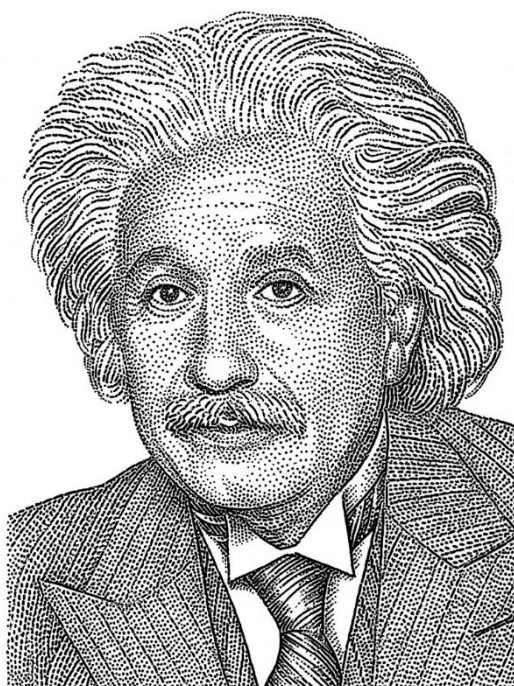
第三步：继续 Google 一张黑白的爱因斯坦。



好了，找到了完美的黑白爱因斯坦。等等，好像哪里不对？这个人好像是教父？继续寻找。

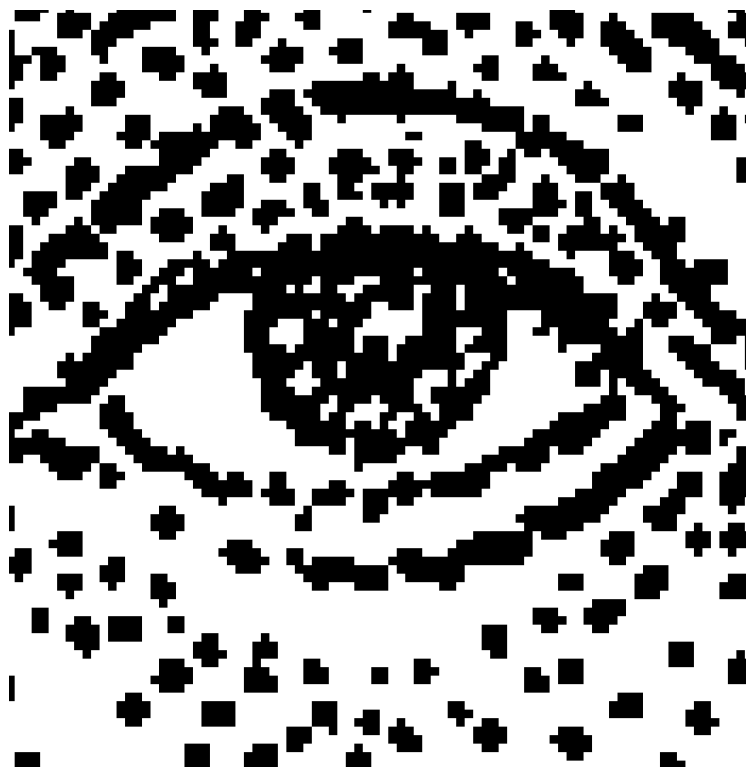


我要是会 PS 就不用这么麻烦了，想想中应该是这样的，可惜不会 P 继续找。

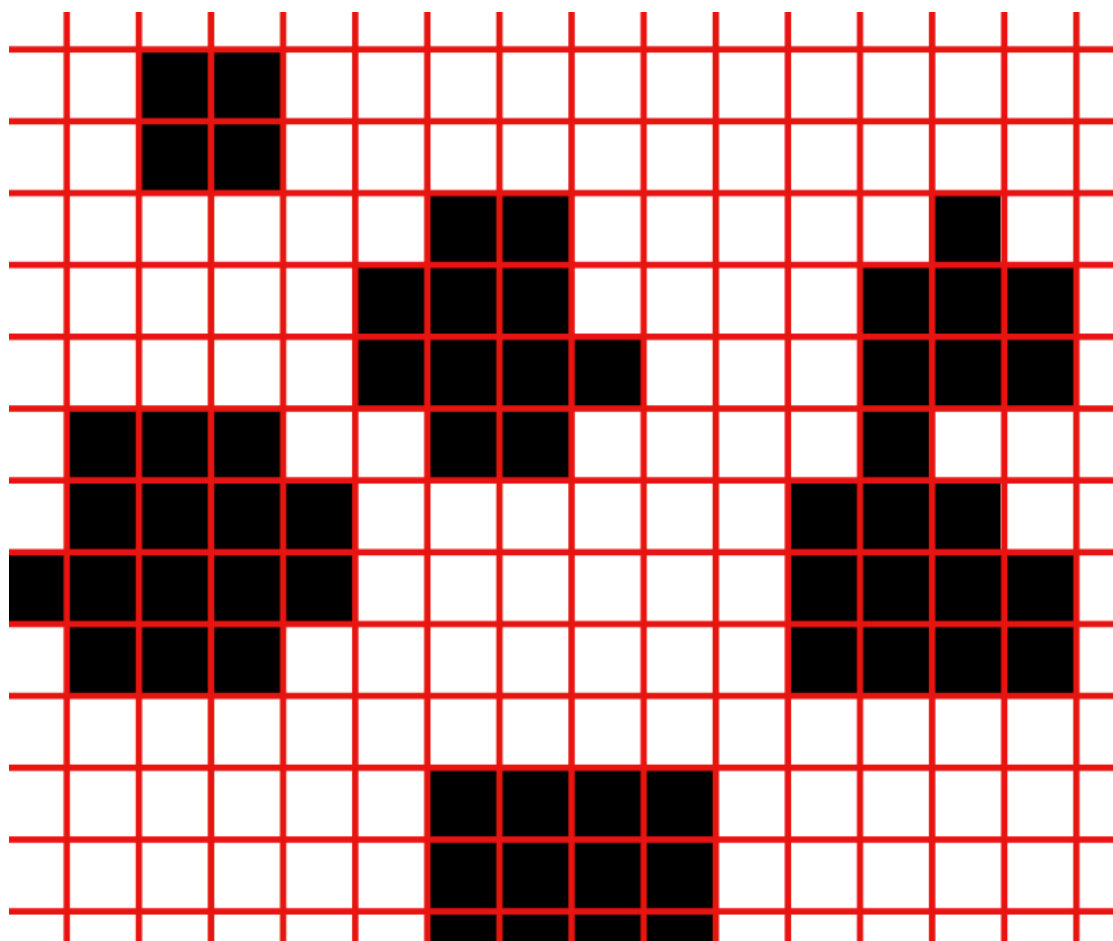


完美地找到了一个黑白的爱因斯坦。

第四步，图片是点阵图，需要转换成矢量图才能绘制。



什么是点阵图，放大了看就是由一个一个点组成的图像。

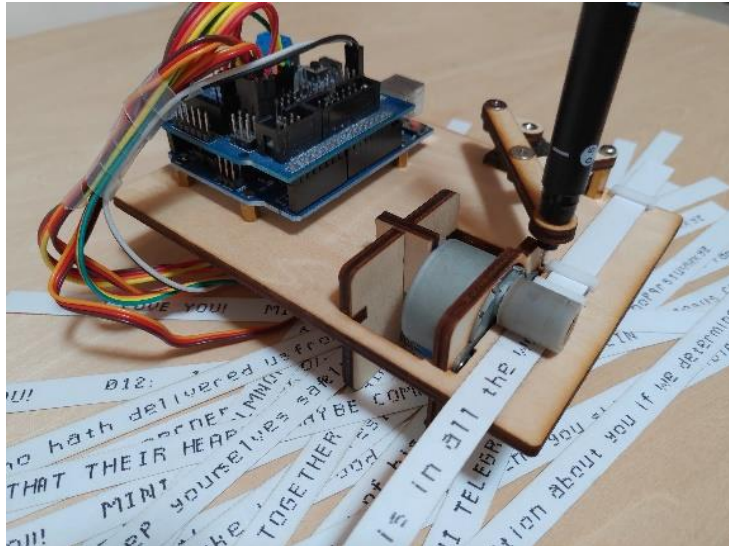


再放大，每个格子是一种颜色，由很多很多个格子组合在一起拼成一个图案。上面这个简

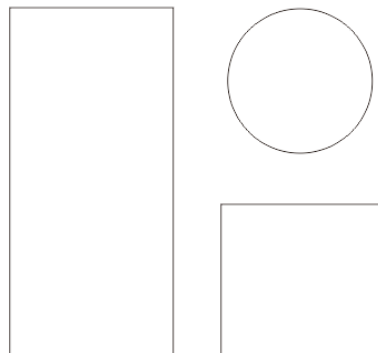
单的爱因斯坦是由横 781*竖 1024 一共 799744 个黑白格子组成的。黑色点大约是 23.1 万个。我们可以知道每个黑色点的位置，然后用笔在纸张上打点就可以了。每个点 1 秒就需要 63 个小时完成，似乎有点慢。

插广告！说到打点可以看看我们店里的电报机

<https://item.taobao.com/item.htm?spm=a1z0d.6639537.1997196601.237.154c7484xanMSf&id=576633734176>，这个才是专业打点的机器。



显然打点不适合我们的拉线墙画机，它是绘制矢量图的神器。那啥是矢量图呢？

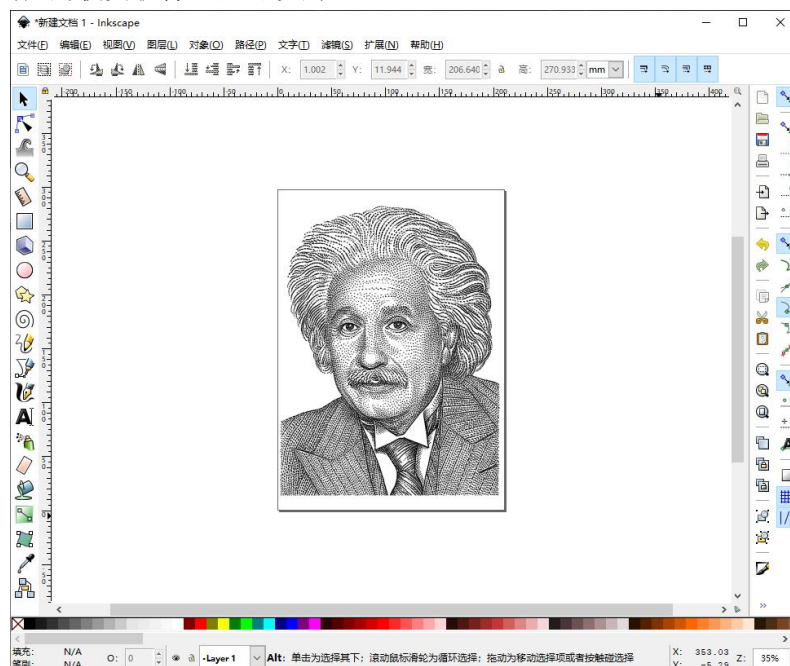


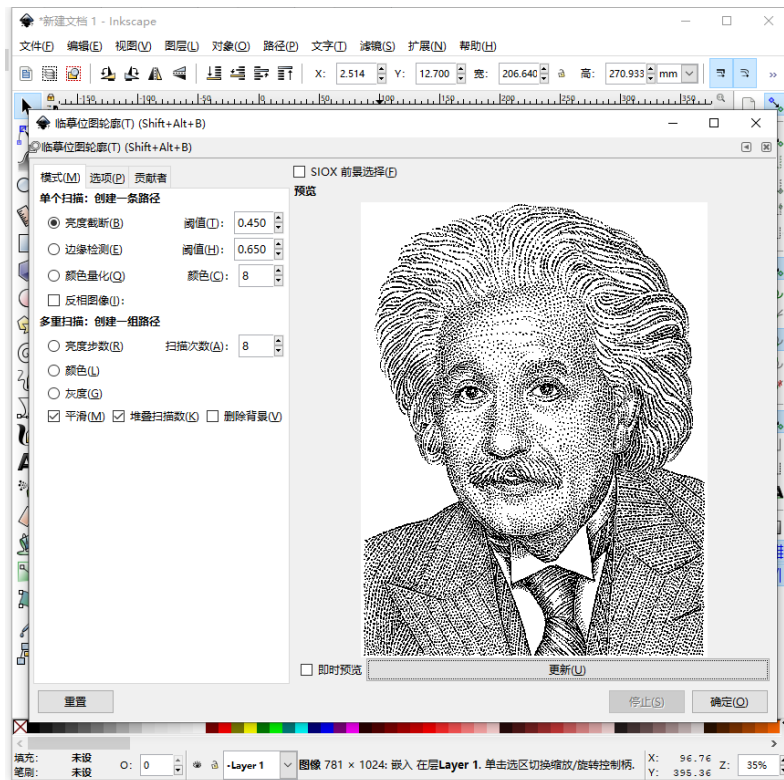
一些图形或者线条，用数字记录该图形的位置，大小，颜色，线条粗细等信息，每次看到的图，都是先计算再绘制出来的图（它们在屏幕上最终还是以点阵图的形式显示出来）。这种图的好处是可以任意放大，无论多大都不会产生点阵图中的那种方块点。坏处是只能表现相对简单的图案。



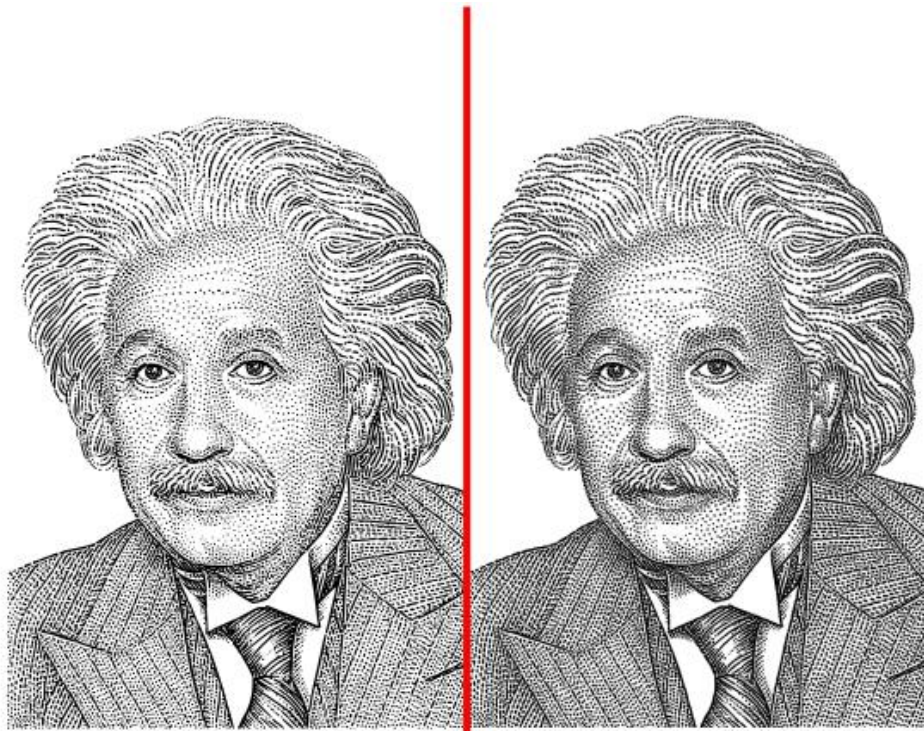
打脸了，这个也是矢量图，一点也不简单。这些都是上过正经美术课（比如只有美术课，没有文化课的美院）的大神画出来的（咱们没上过美术课的就别想了）。补一句，矢量图最擅长曲线，可以秒杀点阵图。上图中每一缕头发都是一个曲线，再加上不同的颜色可以表现出惊人的细节。扯远了。

拉回到爱因斯坦，如何把点阵图变成矢量图呢？答，可以，但不是所有图都可以。有很多种软件都可以实现这个功能。我们讲一个免费软件 **Inkscape**。（你可以到它的官网免费下载使用，当然其他的收费软件也是可以的）

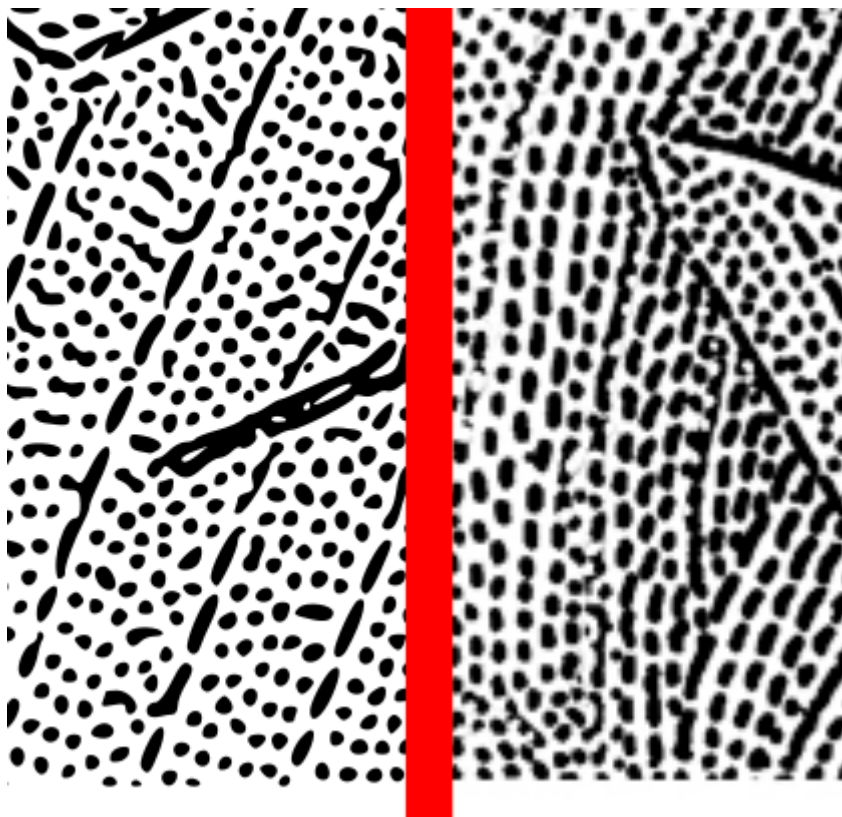




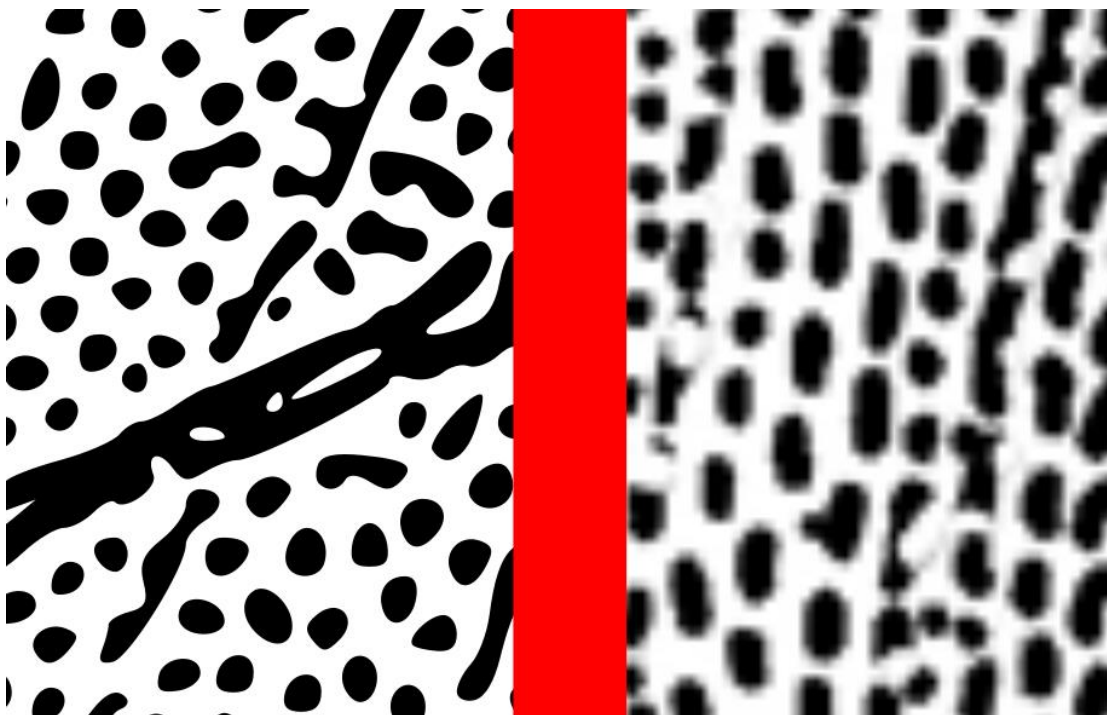
操作很简单，拖爱因斯坦进去，选择爱因斯坦，路径->临摹位图轮廓，确定。



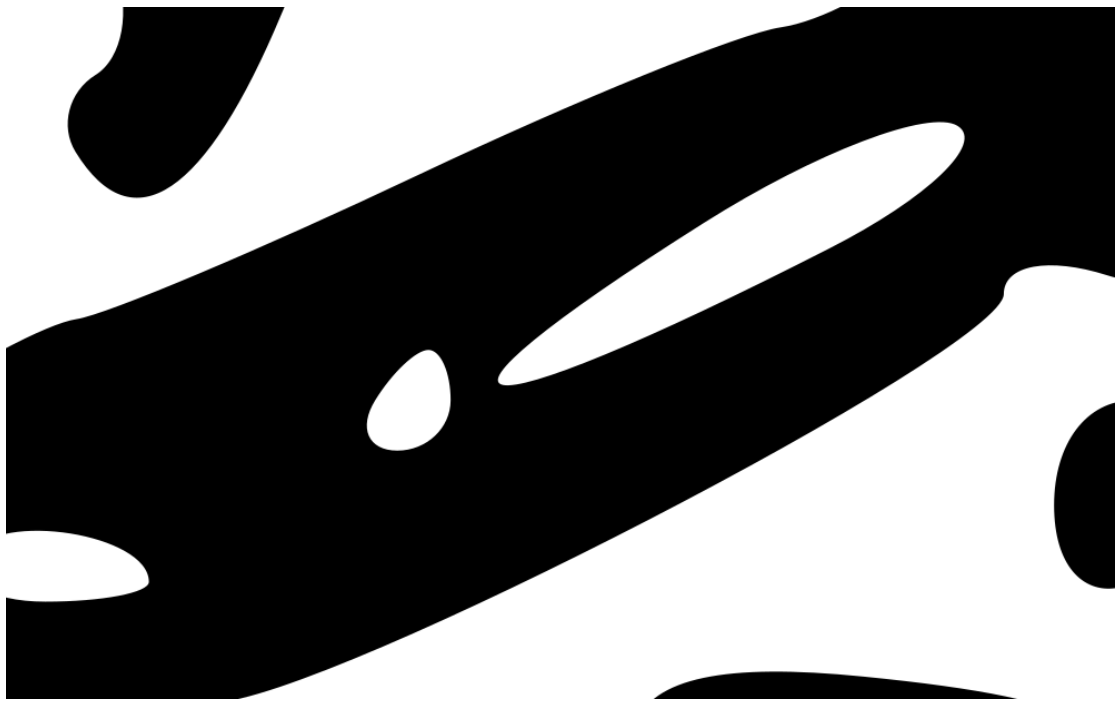
乍一看并不能区分谁是位图，谁是矢量图。



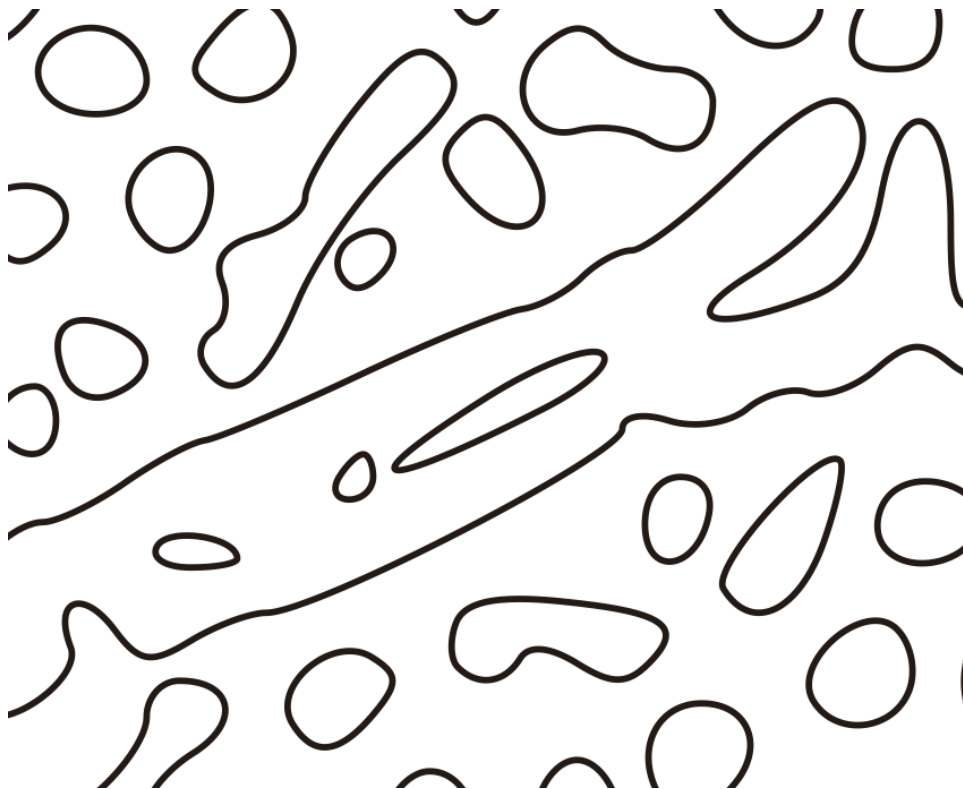
放大到 200%就可以看出一些端倪了。



800%的时候，点阵图已经惨不忍睹了。



矢量图还可以继续毫无压力的放大，放大，再放大。



我们先忽略填色的问题。让笔沿着这些路径走，就可以画好轮廓。要填色的部分，再在里面来回的多走几趟就像小朋友画画那样，先描边，再填色，就搞定了。但我们有强大的设备，可以精准控制，绘制多个轮廓，每次都缩进一个笔道的宽度就可以填满色了。只要让笔沿着我规划的路径移动，就可以画出来了。这些并不重要，只是时间问题。



黑色勾边，黄色涂实就，只要计算好笔道的宽度，就会变成实色的图案了。



可怜我的密集恐惧症啊!!!!

实际上这 2 个步骤没有区别，都是让墙画机走一个特定的路径。所以合并同类项，忽略第二步，减少工作量。我们只先搞一个轮廓就好了。主要是为了我的密集恐惧症否则就不能继续了。



接下来，需要搞一个叫做 **G-code** 的东西（**G-code** 是国际上最广泛使用的数控编程语言），就是将图案中的路径，转换成一大串位置坐标，让机器再沿着这些坐标一点一点移动就可以加工出我们需要的图案了。通俗点说，就是告诉机器从坐标 xy 移动到 $x_1, y_1, x_2, y_2, \dots, x_n, y_n$ 。是不是觉得哪里不对？横平竖直的直线还可以理解，曲线是如何描述的？这里好像又变成了点阵图吧？没错，**G-code** 就是点阵图。只是有小数点而已。在 **G-code** 里几乎没有曲线，任何曲线都被拆分成了小的直线段（避免杠精，有些设备自带曲线功能，只要描述曲线的参数，机器可以自动按曲线运行）。



就连这种德国的 5 轴数控加工中心也是用的 **G-code**，各种小直线！实际上 **G-code** 的计算也是一种非常复杂的处理，他需要计算各种电机的加速，减速，惯性，被加工工件的材质，刀具的尺寸，加工精度，速度等等等等因素。精密的加工中心可以达到微米级的移动控制，都这么高精度了还需要什么曲线。。。

```

5 G01 X0.948 Y-1.592 Z0
7 G01 X0.906 Y-1.219 Z0
3 G01 X0.886 Y-0.850 Z0
9 G01 X0.894 Y-0.669 Z0
0 G01 X0.918 Y-0.490 Z0
1 G01 X1.013 Y0.004 Z0
2 G01 X1.140 Y0.517 Z0
3 G01 X1.220 Y0.748 Z0
4 G01 X1.315 Y0.943 Z0
5 G01 X1.425 Y1.089 Z0
6 G01 X1.487 Y1.140 Z0
7 G01 X1.553 Y1.173 Z0
8 G01 X1.820 Y1.249 Z0
9 G01 X2.105 Y1.298 Z0
0 G01 X2.401 Y1.320 Z0
1 G01 X2.702 Y1.316 Z0
2 G01 X3.002 Y1.286 Z0
3 G01 X3.295 Y1.232 Z0
4 G01 X3.574 Y1.153 Z0
5 G01 X3.834 Y1.051 Z0
6 G01 X3.895 Y1.013 Z0
7 G01 X3.951 Y0.957 Z0
8 G01 X4.049 Y0.801 Z0
9 G01 X4.127 Y0.598 Z0
0 G01 X4.187 Y0.361 Z0
1 G01 X4.229 Y0.103 Z0
2 G01 X4.256 Y-0.160 Z0
3 G01 X4.264 Y-0.650 Z0
4 G01 X4.206 Y-1.713 Z0
5 G01 X4.121 Y-2.776 Z0
6 G01 X3.990 Y-3.830 Z0
7 G01 X3.902 Y-4.352 Z0
8 G01 X3.796 Y-4.869 Z0
9 G01 X3.710 Y-5.162 Z0
0 G01 X3.590 Y-5.448 Z0
1 G01 X3.443 Y-5.728 Z0
2 G01 X3.278 Y-6.004 Z0
3 G01 X2.921 Y-6.549 Z0
4 G01 X2.578 Y-7.097 Z0
5 G01 X2.219 Y-6.558 Z0
6 G01 X1.848 Y-6.021 Z0
7 G01 X1.674 Y-5.749 Z0
8 G01 X1.517 Y-5.472 Z0
9 G01 X1.383 Y-5.188 Z0
0 G01 X1.278 Y-4.895 Z0
1 G01 X1.192 Y-4.543 Z0
2 G01 X1.131 Y-4.185 Z0

```

拉回来，我们到毫米级的精度就 ok 的。（实际上我们的理论分辨率可以

达到 0.027mm，即使实际精度误差 1 个数量级也可以达到 0.27mm 的分辨率相当惊人了）画个爱因斯坦而已，完全势力碾压。

Inkscape 可以将矢量图制作成 G-code 保存，方法很多请 Google 一下很多，实在不行百度也凑合了。此处略。

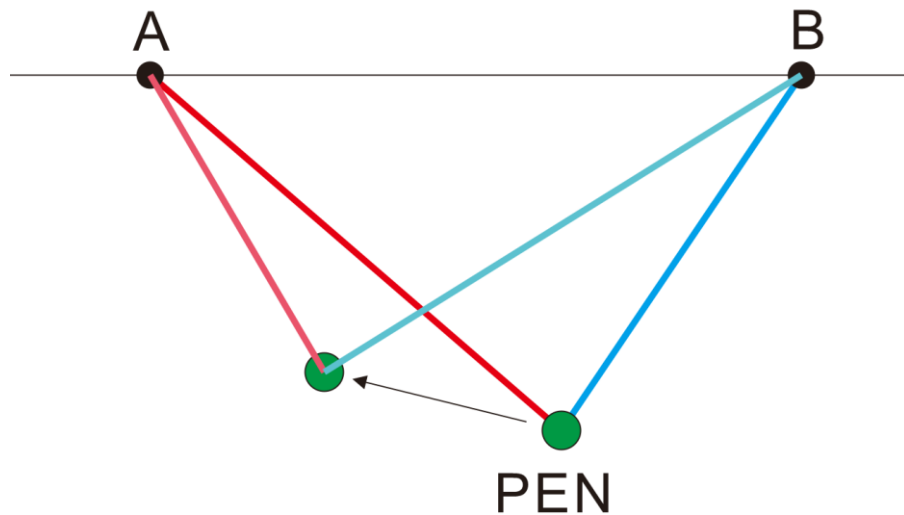
如上我们已经将一张照片，完全变成了，一大串位置坐标。我们只要让设备准确的走到我们指定的位置，再到下一个位置，就 OK 了。

如何走到一个预定的位置呢？我们可以来研究一下主题了。

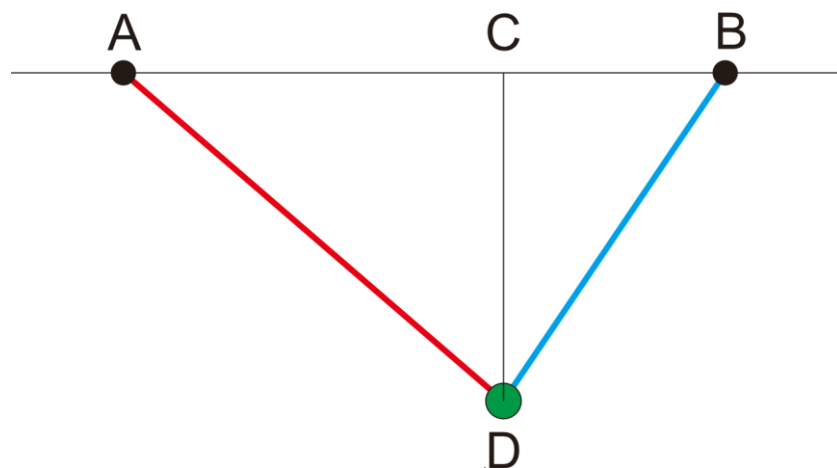


步进电机，请 Google 一下。简单说，是个可以精确控制旋转角度的电机。普通电机通电就转，转了多少，只能估算，不能精确控制，停电后不能 hold 住，有大负载会被反向拉动。而步进电机不但可以精确控制转速和角度（工业级步进电机角度最小可以达到 0.9°，有减速齿轮后可以达到更高精度），而且在静止的时候，还可以有一定的力道保持（不能被外力

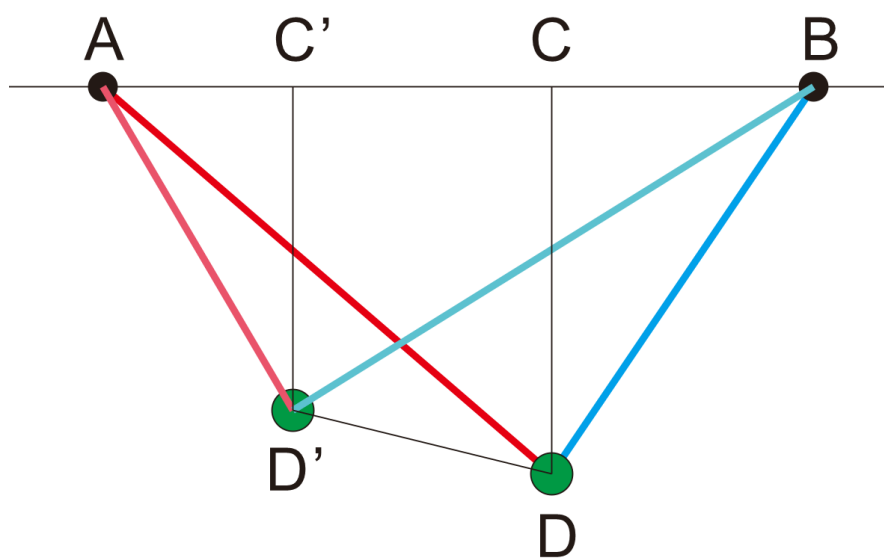
转动)。我们使用了相对好控制的 28BYJ 步进电机，自带减速齿轮，力道可以满足我们的需求。一般空调的导风板都是用这种电机拉动的。接好线我们用程序就可以控制它转，无需额外的驱动器、电源等莫名其妙的设备。更不需要你了解步进电机的细分、距角。。。等莫名其妙的属性。一行代码 `m1.Steps(1)`，转一下。循环 2048 次，转一圈。如果用 35mm 的线轴，转一步，可以拉动 0.027mm 的线。就是这么简单，步进电机就算讲完了。



接下来看看两条线是如何拉动笔尖的。两个步进电机在 AB 点，电机转动线轴，收放两条线，笔就会跟着移动。只要让 A 电机拉线，B 电机放线，笔就可以移动到要的位置。线需要收多少距离，放多少线，也很容易计算，三角函数了解一下？



以知线段 AB，CD 的长度，求 AD，BD？忘了的请找个 14 岁骚年问一问。因为我只记得美术课被数学老师抢了，课上讲的啥也忘了。骚年和我说， $AD = \sqrt{AC^2 + CD^2}$ 勾股定理小学的时候学的，他也快忘记了。



要从 D 移动到 D' 的位置，只需要计算出 AD 和 AD' 的差，BD 和 BD' 的差，再看放多少线，收多少线，算一下电机应该正转还是反转、转多少步就 OK 了。就是这么简单啊。

然后我们只要读取 **G-code**，一行一行的按坐标移动笔，一下一下的就可以把马龙·白兰度的教父画出来了。

