

CSDN

博客 (http://blog.csdn.net?ref=toolbar)学院 (http://edu.csdn.net?ref=toolbar)

(http://www.csdn.net?ref=toolbar)

下载 (http://download.csdn.net?ref=toolbar)更多 ▾

Q

写博客

(http://write.blog.csdn.net/postedit?)

登录 (https://passport.csdn.net/account/login?ref=toolbar)注册 (http://passport.csdn.net/account/mobileregister?ref=toolbar&action=mobileRegister)

图像放缩中最近邻插值和双线性插值的基本原理

转载

2009年11月17日 00:28:00

标签：算法 (http://so.csdn.net/so/search/s.do?q=算法&t=blog) /

工具 (http://so.csdn.net/so/search/s.do?q=工具&t=blog) /

工作 (http://so.csdn.net/so/search/s.do?q=工作&t=blog)

18515

图像的缩放很好理解,就是图像的放大和缩小。传统的绘画工具中,有一种叫做“放大尺”的绘画工具，画家常用它来放大图画。当然，在计算机上，我们不再需要用放大尺去放大或缩小图像了，把这个工作交给程序来完成就可以了。下面来讲讲计算机怎么来放大缩小图象；在本文中，我们所说的图像都是指点阵图，也就是用一个像素矩阵来描述图像的方法，对于另一种图像：用函数来描述图像的矢量图，不在本文讨论之列。

越是简单的模型越适合用来举例子，我们就举个简单的图像：3X3 的256级灰度图，也就是高为3个象素，宽也是3个象素的图像，每个象素的取值可以是 0—255，代表该像素的亮度，255代表最亮，也就是白色，0代表最暗，即黑色。假如图像的象素矩阵如下图所示（这个原始图把它叫做源图，Source）：

234 38 22

67 44 12

89 65 63

这个矩阵中，元素坐标(x,y)是这样确定的，x从左到右，从0开始，y从上到下，也是从零开始，这是图象处理中最常用的坐标系，就是这样一坐标：

----->X

|

|

|

|

|

VY

如果想把这副图放大为 4X4大小的图像，那么该怎么做呢？那么第一步肯定想到的是先把4X4的矩阵先画出来再说，好了矩阵画出来了，如下所示，当然，矩阵的每个像素都是未知数，等待着我们去填充（这个将要被填充的图的做法叫做目标图,Destination）：

? ? ? ?

? ? ? ?

? ? ? ?

? ? ? ?

然后要往这个空的矩阵里面填值了，要填的值从哪里来呢？是从源图中来，好，先填写目标图最左上角的象素，坐标为（0，0），那么该坐标对应源图中的坐标可以由如下公式得出：

srcX=dstX*（srcWidth/dstWidth），srcY = dstY *（srcHeight/dstHeight)

好了，套用公式，就可以找到对应的原图的坐标了(0*(3/4),0*(3/4))=>(0*0.75,0*0.75)=>(0,0)

,找到了源图的对应坐标,就可以把源图中坐标为(0,0)处的234象素值填进去目标图的(0,0)这个位置了。



Andrew659 (http://blog....)

+ 关注

(http://blog.csdn.net/Andrew659)

码云

原创

粉丝

喜欢

未开通 (https://giutm_sourc

82

48

0

- 他的最新文章
- 更多文章 (http://blog.csdn.net/Andrew659)
- Windows使用Computer Modern Roman字体 (/andrew659/article/details/42151665)
- Visio 2010去除页面白边 (/andrew659/article/details/12346073)
- 字符集笔记 (/andrew659/article/details/8571345)
- 简单的command line编译Java Project (/andrew659/article/details/8568338)

在线课程



用户画像系统应用与技术解析

讲师: 刘松林

http://edu.csdn.net/huiyiCourse/series_detail/ce=wx2)



2017龙服面试集训营VIP服务版

讲师: 刘松林

http://edu.csdn.net/huiyiCourse/series_detail/70?utm_source=blog9)

- 热门文章
- Mac OSX上的python (/andrew659/article/details/6103950)
- 21262
- 图像放缩中最近邻插值和双线性插值的基本原理 (/andrew659/article/details/4818988)
- 18495
- Eclipse代码自动补全设置 (/andrew659/article/details/6095234)

接下来,如法炮制,寻找目标图中坐标为(1,0)的像素对应源图中的坐标,套用公式:
(1*0.75,0*0.75)=>(0.75,0)
结果发现,得到的坐标里面竟然有小数,这可怎么办?计算机里的图像可是数字图像,像素就是最小单位了,像素的坐标都是整数,从来没有小数坐标。这时候采用的一种策略就是采用四舍五入的方法（也可以采用直接舍掉小数位的方法），把非整数坐标转换成整数，好，那么按照四舍五入的方法就得到坐标（1，0），完整的运算过程就是这样的：
(1*0.75,0*0.75)=>(0.75,0)=>(1,0)
那么就可以再填一个像素到目标矩阵中了，同样是把源图中坐标为(1,0)处的像素值38填入目标图中的坐标。




依次填完每个像素，一幅放大后的图像就诞生了，像素矩阵如下所示：

```
234  38  22  22
67   44  12  12
89   65  63  63
89   65  63  63
```

这种放大图像的方法叫做最临近插值算法，这是一种最基本、最简单的图像缩放算法，效果也是最不好的，放大后的图像有很严重的马赛克，缩小后的图像有很严重的失真；效果不好的根源就是其简单的最临近插值方法引入了严重的图像失真，比如，当由目标图的坐标反推得到的源图的坐标是一个浮点数的时候，采用了四舍五入的方法，直接采用了和这个浮点数最接近的像素的值，这种方法是很不科学的，当推得坐标值为 0.75的时候，不应该就简单的取为1，既然是0.75，比1要小0.25，比0要大0.75 ,那么目标像素值其实应该根据这个源图中虚拟的点四周的四个真实的点来按照一定的规律计算出来的，这样才能达到更好的缩放效果。双线性内插值算法就是一种比较好的图像缩放算法，它充分的利用了源图中虚拟点四周的四个真实存在的像素值来共同决定目标图中的一个像素值，因此缩放效果比简单的最邻近插值要好很多。双线性内插值算法描述如下：

对于一个目的像素，设置坐标通过反向变换得到的浮点坐标为(i+u,j+v) (其中i、j均为浮点坐标的整数部分，u、v为浮点坐标的小数部分，是取值[0,1)区间的浮点数)，则这个像素得值 f(i+u,j+v) 可由原图像中坐标为 (i,j)、(i+1,j)、(i,j+1)、(i+1,j+1)所对应的周围四个像素的值决定，即：
$$f(i+u,j+v) = (1-u)(1-v)f(i,j) + (1-u)vf(i,j+1) + u(1-v)f(i+1,j) + uvf(i+1,j+1)$$
 公式1
其中f(i,j)表示源图像(i,j)处的像素值，以此类推。

比如，象刚才的例子，现在假如目标图的像素坐标为（1，1），那么反推得到的对应于源图的坐标是（0.75，0.75），这其实只是一个概念上的虚拟像素,实际在源图中并不存在这样一个像素,那么目标图的像素（1，1）的取值不能够由这个虚拟像素来决定，而只能由源图的这四个像素共同决定：（0，0）（0，1）（1，0）（1，1），而由于（0.75,0.75）离（1，1）要更近一些，那么（1,1）所起的决定作用更大一些，这从公式1中的系数uv=0.75×0.75就可以体现出来，而（0.75,0.75）离（0，0）最远，所以（0，0）所起的决定作用就要小一些，公式中系数为(1-u)(1-v)=0.25×0.25也体现出了这一特点

-  14173
- js、html中的单引号、双引号及其转义使用(转)
(/andrew659/article/details/4800756)
-  11890
- win7 ping不通xp问题
(/andrew659/article/details/5352683)
-  11528



-  shiyongraow (/shiyongraow) 2017-09-20 21:38

根据博主给出的3x3矩阵，在matlab中算出来是错误的，就没人去matlab中验证一下吗？？？？！！！！

回复
-  cumtml (/cumtml) 2017-08-25 11:21

用的真好，看懂了双线性插值

回复
-  u010655903 (/u010655903) 2015-01-28 22:44

博主好

回复

查看 10 条热评

相关文章推荐



图像缩放--最近邻插值 (/wmn7q/article/details/52743284)

图像缩放--最近邻插值 自己一直以为放大图像是会放大像素点的大小的，然后就去查了一下，发现不是这样的，而是像素点数量变了，而多的或者少的就依靠插值来实现 百度问答 这里的答案...

 WMN7Q (<http://blog.csdn.net/WMN7Q>) 2016-10-06 13:32  909



图像缩放--OpenCV cvResize函数--最近邻插值---双线性插值--基本原理 (/sfl8688/article/details/8833163)

图像大小变换 void cvResize(const CvArr* src, CvArr* dst, int interpolation=CV_INTER_LINEAR); src输入图像.d...

 sfl8688 (<http://blog.csdn.net/sfl8688>) 2013-04-22 09:42  1013

最临近插值和双线性插值方法 (/qq_19764963/article/details/49074609)

在数字图像处理中，插值方法主要在图像的缩放时候使用。首先使用点阵图观点来看待这个问题，那么插值要解决的就是在新的像素点位置要用什么灰度值的问题。主要分为两步，一是新的点要映射到原图的那个点，当然，映射...

 qq_19764963 (http://blog.csdn.net/qq_19764963) 2015-10-12 15:42  1502


最邻近插值法(The nearest interpolation)实现图像缩放 (/tianxiaov/article/details/51112928)

也称零阶插值。它输出的像素灰度值就等于距离它映射到的位置最近的输入像素的灰度值。但当图像中包含像素之间灰度级有变化的细微结构时,最邻近算法会在图像中产生人为加工的痕迹。 具体计算方法：对于一个目的坐标...

 TianxiaoV (<http://blog.csdn.net/TianxiaoV>) 2016-04-10 14:57  1547

最近邻内插值与双线性插值 (/zhangla1220/article/details/41014541)

最近邻内插值与双线性插值的实现

 zhangla1220 (<http://blog.csdn.net/zhangla1220>) 2014-11-11 17:36  2539

图像内插-最近邻内插法 (/ilyhl5201314/article/details/10713297)

一、图像内插-最近邻内插法 1、数学原理 当一幅二维数字图像从源图像 $N \times M$ 被放为 $(j \times N) \times (k \times M)$ 目标图像是，参照数学斜率计算公式 必然有: $(X1 - Xmin)/...$

 ily6418031hwm (<http://blog.csdn.net/ily6418031hwm>) 2013-08-31 09:12  4722

图像缩放算法及速度优化——(一)最近邻插值 (/willian0621/article/details/8685236)

第0节 简介 图像缩放算法是数字图像处理算法中经常遇到的问题。我们经常会将某种尺寸的图像转换为其他尺寸的图像，如放大或者缩小图像。OpenCV中的Resize() 函数非常方便而且效率非常...

 willian0621 (<http://blog.csdn.net/willian0621>) 2013-03-17 22:56  7894

程序员必读经典书籍 （转） (/zhangla1220/article/details/42522157)

原文地址：<http://iteye.blog.163.com/blog/static/186308096201271931628953/> 1、《代码大全》 史蒂夫·迈克康奈尔 ...

 zhangla1220 (<http://blog.csdn.net/zhangla1220>) 2015-01-08 11:19  558


图像缩放算法及速度优化——(一)最近邻插值 (/lvyanming/article/details/48163063)

第0节 简介 图像缩放算法是数字图像处理算法中经常遇到的问题。我们经常会将某种尺寸的图像转换为其他尺寸的图像，如放大或者缩小图像。OpenCV中的Resize() 函数非常方便而且效率非常...


 lvyanming (<http://blog.csdn.net/lvyanming>) 2015-09-01 20:34  1171

图像放缩中最近邻插值和双线性插值的基本原理 (/lzhf1122/article/details/72510873)

图像的缩放很好理解,就是图像的放大和缩小。传统的绘画工具中,有一种叫做“放大尺”的绘画工具,画家常用它来放大图画。当然,在计算机上,我们不再需要用放大尺去放大或缩小图像了,把这个工作交给程序来完成就可...


 lzhf1122 (<http://blog.csdn.net/lzhf1122>)

2017-05-18 22:08


 78

图像缩放--OpenCV cvResize函数--最近邻插值---双线性插值--基本原理 (/u012900447/article/details/53188053)

图像大小变换 void cvResize(const CvArr* src, CvArr* dst, int interpolation=CV_INTER_LINEAR); src输入图像.ds...


 u012900447 (<http://blog.csdn.net/u012900447>)

2016-11-16 16:34


 982

图像缩放--OpenCV cvResize函数--最近邻插值---双线性插值--基本原理 (/wishlifehappy/article/details/8783290)

图像大小变换 来源: <http://blog.chinaunix.net/uid-26020768-id-3187769.html> void cvResize(const CvArr* sr...


 WishLifeHappy (<http://blog.csdn.net/WishLifeHappy>)

2013-04-10 15:36


 1211


数字图像缩放之最近邻插值与双线性插值处理效果对比 (/yi_tech_blog/article/details/52997234)

基本原理: 1、最近邻插值: 变换后的目标图像某点像素值等于源图像中与变换前相应点最近的点的像素值。具体操作为, 设水平方向和垂直方向缩放的比例分别为w和h, 那么目标图像中的点des(x,y)对...

 yi_tech_blog (http://blog.csdn.net/yi_tech_blog)

2016-11-01 16:32


 1300




<http://download.csdn.net/detail/anywang1011562680/4118033>

自编的可放大任意整数倍的最近邻插值和双线性插值函数 (http://download.csdn.net/detail/sinat_36412790/article/details/75257954)

最近邻插值公式如下: $D(x,y) = \text{最近邻插值} * \text{权重系数}$ 输入: src:输入源图 dst_rows:目标图的行数 dst_cols:目标图的列数 输出: 目标图 */ Mat Nearge...


 sinat_36412790 (http://blog.csdn.net/sinat_36412790)

2017-07-17 17:29


 105

图像放缩之双线性内插值 (/jia20003/article/details/6915185)

一: 数学原理 在临近点插值的数学基础上, 双线性插值, 不是简单copy源像素的值, 而是获取四个最邻近目标像素的像素值乘以权重系数, 简单的数学公式可以表示为: $D(x, y) = S(j, k) \dots$

 jia20003 (<http://blog.csdn.net/jia20003>)

2011-10-28 17:03

 19348




<http://download.csdn.net/detail/shuchengzhang/6540557>


图像放缩之双线性内插值 (<http://download.csdn.net/detail/shuchengzhang/6540557>)

图像放缩之双线性内插值 (/baidu_22742607/article/details/45498625)

图像放缩之双线性内插值: 在临近点插值的数学基础上, 双线性插值, 不是简单copy源像素的值, 而是获取四个最邻近目标像素的像素值乘以权重系数, 简单的数学公式可以表示为: $D(x, y) = \dots$

 baidu_22742607 (http://blog.csdn.net/baidu_22742607)

2015-05-05 09:45

 254

双线性插值进行图像缩放 (http://download.csdn.net/detail/qq_19944227/9933387)



2017-08-14 20:17 2KB [下载](#)

(<http://download>
