**数字图像处理课程作业（一）**

**题目(中) 数字图像基础操作**

**姓名与学号 XXX XXXXXXXX**

**年级与专业 XXXXXXXXX**

**所在学院 XXXXXXXXXXXXX**

**二〇一八年十一月**

目录

[第一章 基本原理 5](#_Toc529800377)

[1 数字图像表示 5](#_Toc529800378)

[2 双线性内插法 5](#_Toc529800379)

[第二章 代码实现 6](#_Toc529800380)

[1 双线性内插法 6](#_Toc529800381)

[2 测试脚本 7](#_Toc529800382)

[3 实现效果 8](#_Toc529800383)

[第三章 课程作业 10](#_Toc529800384)

[1.1 编程实现双线性插值 10](#_Toc529800385)

[1.2 实现图像压缩 10](#_Toc529800386)

[1.3 实现图像放大 10](#_Toc529800387)

[参考文献 11](#_Toc529800388)

[附件 12](#_Toc529800389)

图像目录

[Figure 1 原始图像 8](#_Toc529800390)

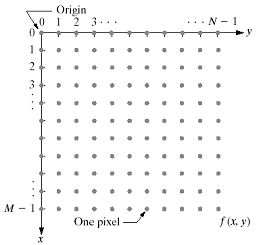
[Figure 2 双线性内插法缩小图像 9](#_Toc529800391)

[Figure 3 双线性内插法放大图像 9](#_Toc529800392)

# 第一章 基本原理

## 1 数字图像表示

一副灰度数字图像由个像素点构成，每个像素点在计算机中占k个二进制位，则一副图像所需的存储空间为比特。一般情况，也就是说每个像素点由一个字节表示，取值范围为共256个灰度级，其中0表示黑色，255表示白色。以图像左上角为原点建立坐标系，向下为x轴，向右为y轴，可表示为如下的坐标形式：



也可表示为矩阵形式：

在matlab中，下标从1开始。

## 2 双线性内插法

双线性内插法是一种数字图像缩放方法。如果要将一副的图像放大到，可以先在原来的图像栅格上画出栅格，对于栅格上每个像素点，根据原栅格上的附近像素点的值计算出新灰度值填入。如果新图像的像素值由原图像栅格上最靠近的像素点灰度值赋值得到，这种方法称为最近领域内插法。最近领域内插法速度快，但是会产生马赛克效应。

双线性内插法是对最近邻域内插法的改进，即新像素的值由原图像上最邻近的4个像素点经两次线性计算得到。



设原图像为，新图像上点对应原图像上的点为，其中，分别为原图像坐标的整数部分和小数部分。可由公式计算得到：

# 第二章 代码实现

## 1 双线性内插法

编写一个实现双线性内插法的函数bilinear.m，输入一幅图像矩阵及目标图像的大小（行、列数），输出一幅缩放后的图像。

function dst = bilinear(src, row, col)

%BILINEAR bilinear interpolation

%

% INPUT

% SRC source image

% ROW,COL size of destination image

% OUTPUT

% DST destination image

% get the size of source image

[m,n] = size(src);

% calculate row and column steps

rstep = (m-1)/(row-1);

cstep = (n-1)/(col-1);

% initialize destination image

dst = zeros(row, col);

% in case of index out of range

src(m+1,:)=0;

src(:,n+1)=0;

% calculate destination image pixel by pixel

for i = 1:row

for j = 1:col

x = 1+rstep\*(i-1);

y = 1+cstep\*(j-1);

% fractional parts

u = x-floor(x);

v = y-floor(y);

% integer parts

x = floor(x);

y = floor(y);

dst(i,j) = (1-u)\*(1-v)\*src(x,y) +...

(1-u)\*v\*src(x,y+1) +...

u\*(1-v)\*src(x+1,y) +...

u\*v\*src(x+1,y+1);

end

end

end

## 2 测试脚本

编写一个测试脚本test.m以测试双线性内插法的实现效果。

clear;clc;close all;

img = imread('2.Fig2.19(a).jpg');

shrink = uint8(bilinear(img, 256, 256));

zoom = uint8(bilinear(shrink, 1024,1024));

imwrite(img,'images/source.jpg');

imshow(img);

title('source image');

imwrite(shrink,'images/shrink.jpg');

figure;

imshow(shrink);

title('shrink image');

imwrite(zoom,'images/zoom.jpg');

figure;

imshow(zoom);

title('zoom image');

## 3 实现效果



Figure 原始图像



Figure 双线性内插法缩小图像



Figure 双线性内插法放大图像

# 第三章 课程作业

## 1 编程实现双线性插值

(a)Write a computer program capable of zooming and shrinking an image by bilinear interpolation. The input to your program is the desired size of the resulting image in the horizontal and vertical direction. You may ignore aliasing effects.

见第二章双线性内插法代码实现。

## 2 实现图像压缩

(b)Download Fig. 2.19(a) and use your program to shrink this image from 1024 x 1024 to 256 x 256 pixels.

见第二章实现效果，双线性内插法缩小图像。

## 3 实现图像放大

(c)Use your program to zoom the image in (b) back to 1024 x 1024.  Explain the reasons for their differences.

见第二章实现效果，双线性内插法放大图像。

图像经双线性内插法缩小后再放大至原图像大小，会出现模糊现象。原因是图像缩小后，部分像素点信息丢失了，再放大后出现的新像素点是经过拟合得到的，不可能完全还原原始图像，因此会出现模糊。另一方面，由于双线性内插法充分利用了4个邻域点信息，能够避免最近邻域内插法造成的马赛克效应。

# 参考文献

1. 数字图像处理（第三版）。Rafael C.Gonzalez & Richard E. Woods。电子工业出版社

# 附件

1. Matlab文件：
   1. bilinear.m —— 实现双线性内插法的函数
   2. test.m —— 实现图像缩放的可执行脚本
2. 原始图像文件：
   1. 2.Fig2.19(a).jpg
3. 输出图像images
   1. source.jpg —— 原始图像
   2. shrink.jpg —— 缩小图像
   3. zoom.jpg —— 放大图像