实验一 Matlab图像操作基础及图像灰度变换

计算成像与统计学习 5天前

一、 实验目的

了解Matlab编程环境，掌握Matlab中图像的表示方法，图像类型、数据类型的种类及各自的特点，并知道怎样在它们之间进行转换。熟悉Matlab中的DIP (Digital Image Processing)工具箱；掌握Matlab环境下的一些最基本的图像处理操作，如读图像、写图像、查看图像信息和格式、尺寸和灰度的伸缩等等；通过实验掌握图像直方图的描绘方法，加深直方图形状与图像特征间关系间的理解；加深对直方图均衡算法的理解。

二、 实验内容

1． 从目录中读取一幅灰度图像；

2． 显示图像信息，查看图像格式、大小、位深等内容；

3． 通过图像的位平面切割（分解），了解数字图像的计算机表示。

4． 编写求图像直方图的Matlab程序，并画图；

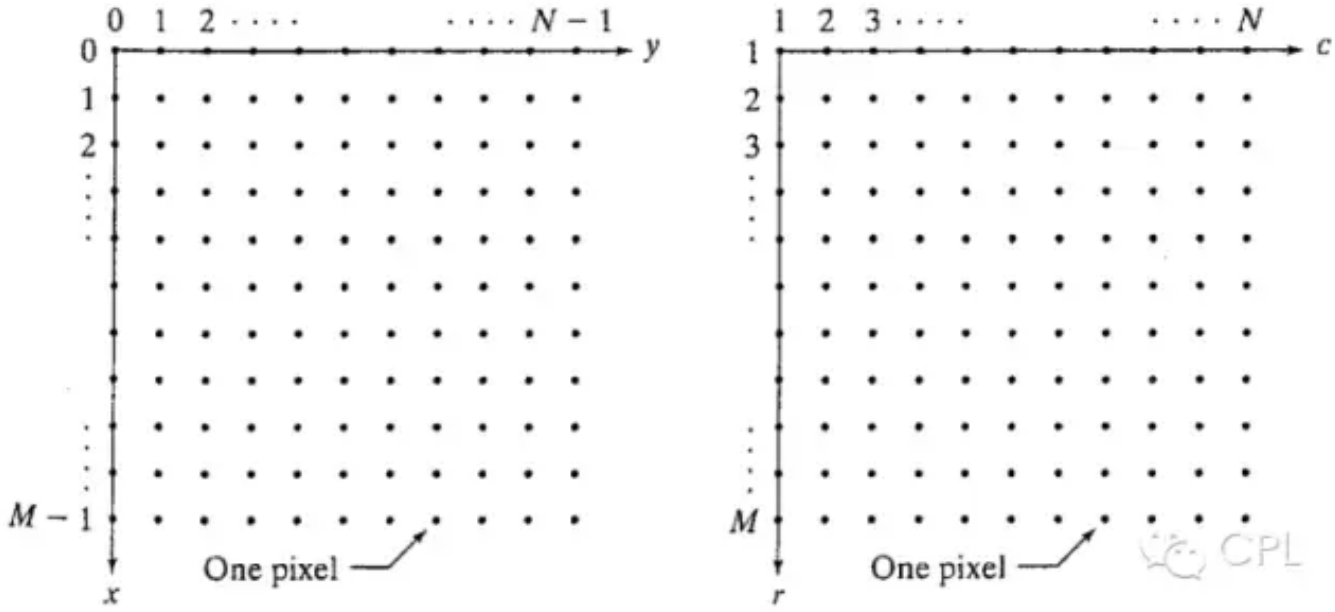
5． 把第3步的结果与直接用Matlab工具箱中函数histogram的结果进行比较，以衡量第3步中程序的正确性。

6． 对读入的图像进行直方图均衡化，画出处理后的直方图，并比较处理前后图像效果的变化。

7． 编写求直方图均衡化转换函数的Matlab程序。

三、知识要点

1. Matlab 坐标规定



图像的矩阵表达形式：

$$f(x,y)=\begin{pmatrix} f(0,0), & f(0,1), & \cdots, & f(0,N-1) \\ f(1,0), & f(1,1), & \cdots, & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0), & f(M-1,1), & \cdots, & f(M-1,N-1) \end{pmatrix}$$

2. Matlab 6.5支持的图像图形格式

TIFF, JEPG, GIF, BMP, PNG, XWD (XWindow Dump)，其中GIF不支持写。

Format Name	Description	Recognized Extensions
TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photographic Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format†	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

3. 与图像处理相关的最基本函数

读：**imread**; 写：**imwrite**; 显示：**imshow**; 信息查看：**imfinfo**;

4.Matlab6.5支持的数据类

double, unit8, int8, uint16,int16, uint32, int32, single, char (2 bytes per element), logical.

5. Matlab6.5支持的图像类型

double, unit8, int8, uint16,int16, uint32, int32, single, char (2 bytes per element), logical.

Data type	Description	Range
int8	8-bit integer	−128 — 127
uint8	8-bit unsigned integer	0 — 255
int16	16-bit integer	−32768 — 32767
uint16	16-bit unsigned integer	0 — 65535
double	Double precision real number	Machine specific

6. 数据类及图像类型间的基本转换函数

Intensity images, binary images, indexed images, RGB image

数据类转换：**B = data_class_name(A);**

IPT图像数据类型转换函数：

函数名	输出图像数据类型	输入图像数据类型
im2uint8	uint8	logical, uint8, uint16, double
im2uint16	uint16	logical, uint8, uint16, double
mat2gray	Double ([0, 1])	double
im2doule	double	logical, uint8, uint16, double
im2bw	logical	logical, uint8, uint16, double

四、参考程序和参考结果

1. 图像基本操作

```
%图像读取
f=imread ('Fig0117(b).tif')
f=imread ('Fig0117(b).tif');    %分号的作用

%图像大小
size(f)

%图像的信息
whos f

%图像的详细及信息，imfinfo 显示的信息比 whos 要多
imfinfo ('Fig0117(b).tif')
imfinfo('cameraman.tif')    %可以试着加个“；”结尾试试。

%图像的显示
imshow(f)

help imshow

cd=double(f);
imshow(f), figure, imshow(cd),
figure, imshow(cd/255);
figure, imshow(cd/512);
figure, imshow(cd/128);          %比较一下结果，为什么？

cd=im2double(f);
figure, imshow(cd)

c = [1 2 3
  4 5 6
  7 8 9]
figure, image(c)
image(c), truesize, axis off

%图像的写
imwrite (f, 'MRI.bmp', 'bmp')

help imwrite
```

```
%彩色图像，与灰度图像比较一下
lena=imread('lena_color_512.tif');
whos
imfinfo ('lena_color_512.tif ')
figure, imshow(lena);

%数据类型及转换
a1=5.0123456789
uint8(a1)
uint8(512.12345678910121)

A1 = [0,1,2;3,4,5;6,7,8]
B1=double(A1)
F1=double(f);
F=im2double(f);
figure, imshow(F);
figure, imshow(f);

A2 = [ -0.5 0.5
       0.75 1.5]
G=im2uint8(A2)

A3 = [128, 300; -12,66.98]
G2=mat2gray(A3,[0,255])    %结果与下一行比较
G3=mat2gray(A3)             %结果与上一行比较
%向量操作
V = [1 2 3 4]
w=v'
w (1: end)
zeros (1,4)
ones (2,4)

矩阵的索引
A4 = [1 2 3;4 5 6]
A5 = rand (4,4)
A6=magic(5)          %或者 A6=magic(3)
A=5* ones (3,3)
A5(1,2)
A5(:,2:4)            %或者 A5=(1:3,:)
A7 = [0 1 2 3
   2 3 4 5
   6 7 8 9]

sum(A7)
sum(A7(:))      %也可以单独试一下 A7(:)
sum (A7')
```

```
max(A7)
max (A7(:))
max (A7 ')

min(A7)
min (A7(:))
min (A7 ')

%矩阵大小
size(A7)
numel(A7)

reshape (A7, 1, numel (A7))
reshape (A7, 2, numel(A7)/2)

%保存和导入数据
save work.mat
help save

clear all          %清空所有数据
close all          %关闭所有窗口
load work.mat   %重新导入刚保存的数据
% 导入 matlab 自带的一些数据
load mri;
figure,
montage (D, map)
title ('Horizontal Slices');

%Matlab 中自带的断层切片模拟图像函数，例如
P = phantom ('Modified Shepp-Logan',200);
imshow(P)
ph = phantom (256);
figure, imshow(ph)
```

2. 图像的位平面切割（分解）

```
c=imread('cameraman.tif');

cd=double(c);
c0=mod(cd,2);
c1=mod(floor(cd/2),2);
c2=mod(floor(cd/4),2);
c3=mod(floor(cd/8),2);
c4=mod(floor(cd/16),2);
c5=mod(floor(cd/32),2);
c6=mod(floor(cd/64),2);
c7=mod(floor(cd/128),2);

figure, imshow('c7');          % 同样可以考察一下 c6, c5,…, c0
```

3．求灰度直方图

```
% Experiment 1: calculate the histogram of gray-scale through gray-scale area.
% function .
f=imread('rice4.tif');
[m, n] = size(f);
myhist=zeros (1,256);
% compute the area under certain gray level.
for k=0:255
    ind = find(f == k);
    myhist(k+1) = length(ind);
end
.
figure,
.
subplot (131); imshow(f);
title ('The rice image');
.
% 也可以试一下 myhist=myhist./numel(f);
.
Subplot (132); stem ([1:1:256], myhist, '.');
axis ([1 256 0 max(myhist)]);
title ('Histograms by myhist');
.
subplot (133); imhist(f);
axis ([1 256 0 max(imhist(f))]);
title ('Histograms by" imhist" function');
```

运行结果：

注意与 matlab 自带的 imhist ( ) 函数的运行结果相比较。

直方图均衡化

```
f = imread('rice4.tif');
subplot (221); imshow(f);
title ('the orignal image');
subplot (222); imhist(f);
ylim ('auto');
g = histeq (f, 256);
subplot (223); imshow(g);
title ('image after equalization');
subplot (224); imhist(g);
ylim ('auto');
```

运行结果：

注意比较直方图形态以及图像的视觉效果变化

绘制直方图均衡化的转换函数：

```
I=imread('rice4.tif');
figure,
hnorm=imhist(I). /numel(I);
cdf=cumsum (hnorm);
x = linspace (0,1,256);
plot (x, cdf);
axis ([0 1 0 1]);
set(gca,'xtick',0:.2:1);
set(gca,'ytick',0:.2:1);
xlabel ('Input Intensity Values', 'fontsize',9)
ylabel ('Output Intensity Values', 'fontsize',9)
text (0.18,0.5,'Transformation Function','fontsize',9);
```

运行结果：

注意观察不同图像直方图均匀化的转换函数之间的形状差异。

Operator	Name	MATLAB Function	Comments and Examples
+	Array and matrix addition	plus(A, B)	a + b, A + B, or a + A.
–	Array and matrix subtraction	minus(A, B)	a – b, A – B, A – a, or a – A.
.*	Array multiplication	times(A, B)	C = A.*B, C(I, J) = A(I, J)*B(I, J).
*	Matrix multiplication	mtimes(A, B)	A*B, standard matrix multiplication, or a*A, multiplication of a scalar times all elements of A.
./	Array right division	rdivide(A, B)	C = A./B, C(I, J) = A(I, J)/B(I, J).
.\	Array left division	ldivide(A, B)	C = A.\B, C(I, J) = B(I, J)/A(I, J).
/	Matrix right division	mrdivide(A, B)	A/B is roughly the same as A*inv(B), depending on computational accuracy.
\	Matrix left division	mldivide(A, B)	A\B is roughly the same as inv(A)*B, depending on computational accuracy.
.^	Array power	power(A, B)	If C = A.^B, then C(I, J) = A(I, J)^B(I, J).
^	Matrix power	mpower(A, B)	See online help for a discussion of this operator.
.'	Vector and matrix transpose	transpose(A)	A.'. Standard vector and matrix transpose.
'	Vector and matrix complex conjugate transpose	ctranspose(A)	A'. Standard vector and matrix conjugate transpose. When A is real A.' = A'.
+	Unary plus	uplus (A)	+A is the same as 0 + A.
–	Unary minus	uminus (A)	–A is the same as 0 – A or –1*A.
:	Colon		Discussed in Section 2.8.

Function	Description
imadd	Adds two images; or adds a constant to an image.
imsubtract	Subtracts two images; or subtracts a constant from an image.
immultiply	Multiplies two images, where the multiplication is carried out between pairs of corresponding image elements; or multiplies a constant times an image.
imdivide	Divides two images, where the division is carried out between pairs of corresponding image elements; or divides an image by a constant.
imabsdiff	Computes the absolute difference between two images.
imcomplement	Complements an image. See Section 3.2.1.
imlincomb	Computes a linear combination of two or more images. See Section 5.3.1 for an example.

Operator	Name
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
==	Equal to
~=	Not equal to

Operator	Name
&	AND
|	OR
~	NOT

Function	Comments
xor (exclusive OR)	The xor function returns a 1 only if both operands are logically different; otherwise xor returns a 0.
all	The all function returns a 1 if all the elements in a vector are nonzero; otherwise all returns a 0. This function operates columnwise on matrices.
any	The any function returns a 1 if any of the elements in a vector is nonzero; otherwise any returns a 0. This function operates columnwise on matrices.

Function	Description
iscell(C)	True if C is a cell array.
iscellstr(s)	True if s is a cell array of strings.
ischar(s)	True if s is a character string.
isempty(A)	True if A is the empty array, [ ].
isequal(A, B)	True if A and B have identical elements and dimensions.
isfield(S, 'name')	True if 'name' is a field of structure S.
isfinite(A)	True in the locations of array A that are finite.
isinf(A)	True in the locations of array A that are infinite.
isletter(A)	True in the locations of A that are letters of the alphabet.
islogical(A)	True if A is a logical array.
ismember(A, B)	True in locations where elements of A are also in B.
isnan(A)	True in the locations of A that are NaNs (see Table 2.10 for a definition of NaN).
isnumeric(A)	True if A is a numeric array.
isprime(A)	True in locations of A that are prime numbers.
isreal(A)	True if the elements of A have no imaginary parts.
isspace(A)	True at locations where the elements of A are whitespace characters.
issparse(A)	True if A is a sparse matrix.
isstruct(S)	True if S is a structure.

Function	Value Returned
`ans`	Most recent answer (variable). If no output variable is assigned to an expression, MATLAB automatically stores the result in `ans`.
`eps`	Floating-point relative accuracy. This is the distance between 1.0 and the next largest number representable using double-precision floating point.
`i` (or `j`)	Imaginary unit, as in `1 + 2i`.
`NaN` or `nan`	Stands for Not-a-Number (e.g., `0/0`).
`pi`	3.14159265358979
`realmax`	The largest floating-point number that your computer can represent.
`realmin`	The smallest floating-point number that your computer can represent.
`computer`	Your computer type.
`version`	MATLAB version string.