江西理工大學 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

Digital Image Processing
数字图像处理

Lecture 015:
MATLAB and image Translation

Dr Ata Jahangir Moshayedi

Prof Associate ,
School of information engineering Jiangxi university of science and technology, China

EMAIL: ajm@jxust.edu.cn

Autumn _2021

江西理工大學 信息工程学院
JIANGXI UNIVERSITY OF SCIENCE AND TECHNOLOGY SCHOOL OF INFORMATION ENGINEERING

Digital Image Processing

LECTURE 015: MATLAB and image Translation

Agenda

- **Translation**
  - **Image Scaling**
  - **Image Shearing**
  - **Image Reflection**
  - **Image Rotation**
  - **Image Cropping**

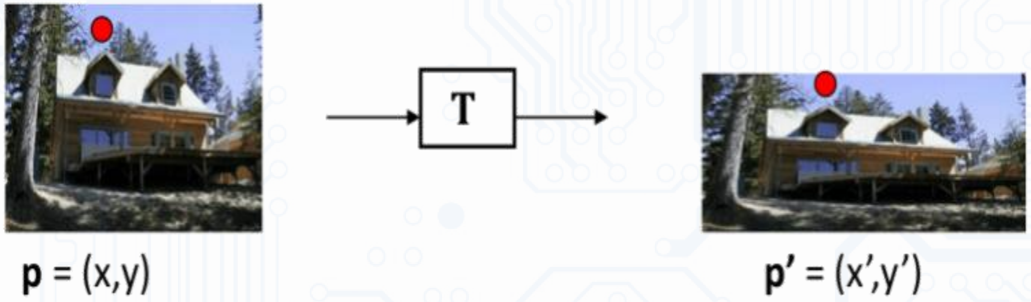
变换：

- 图像缩放。
- 图像剪切。
- 图像反射。
- 图像旋转。
- 图像裁剪。

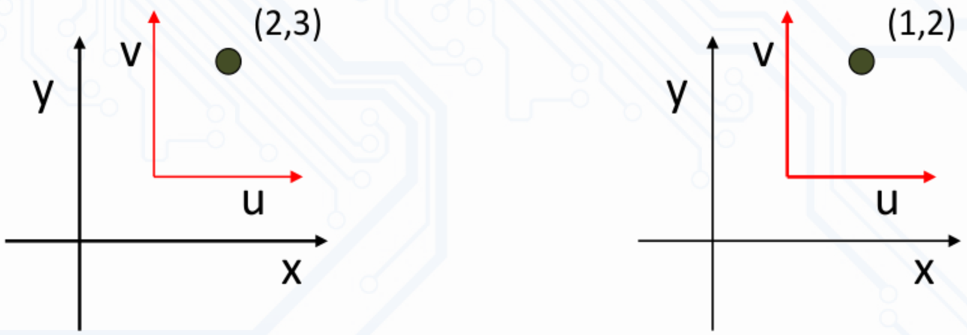
Introduction 介绍

图像变换

Image transformation is a coordinate changing function, it maps some (x, y) points in one coordinate system to points (x', y') in another coordinate system.

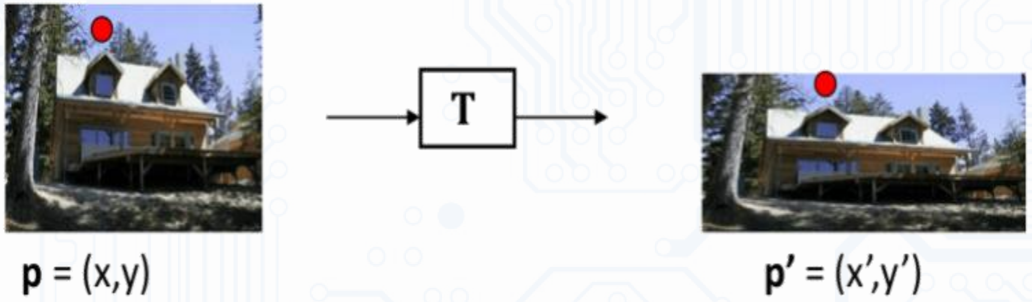


For example, if we have (2, 3) points in x-y coordinate, and we plot the same point in u-v coordinate, the same point is represented in different ways, as shown in the figure below:

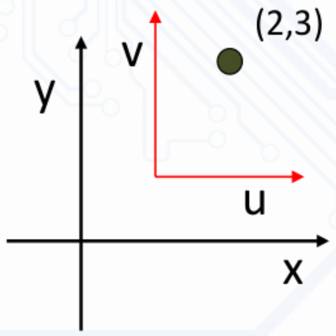


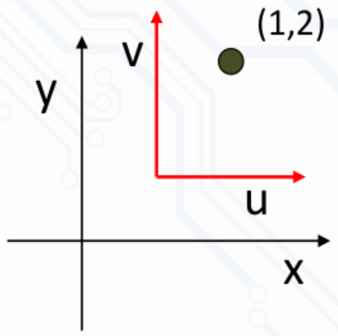
介绍

图像变换是一个坐标变换函数，它将一个坐标系中的(x, y)点映射到另一个坐标系中的(x'， y')点。



例如，在x-y坐标上有(2,3)个点，在u-v坐标上绘制同一点，同一点有不同的表示方式，如下图所示:

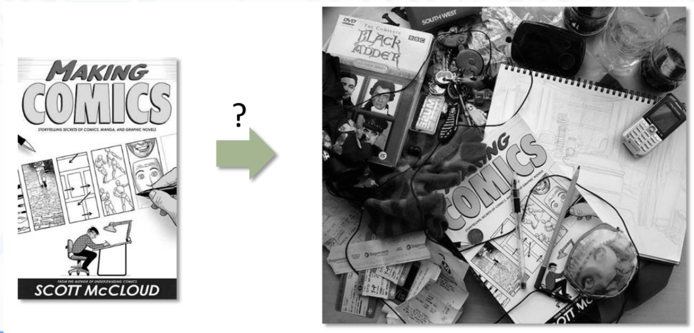




The Use of Image Transformation

图像变换的使用

- In the image below, the geometric relation between the comic book and the image on the right side is based on the similarity transformation (rotation, translation and scaling).
- If we need to train a machine learning model that finds this comic book, then we need to input the image in a different shape and angle.
- Matrices can represent images. Each value in a matrix is a pixel value at a specific coordinate.
- Image transformation can be performed using matrix multiplication. Mathematicians have worked out some matrices that can be used to accomplish certain transformation operations.



图像变换的使用

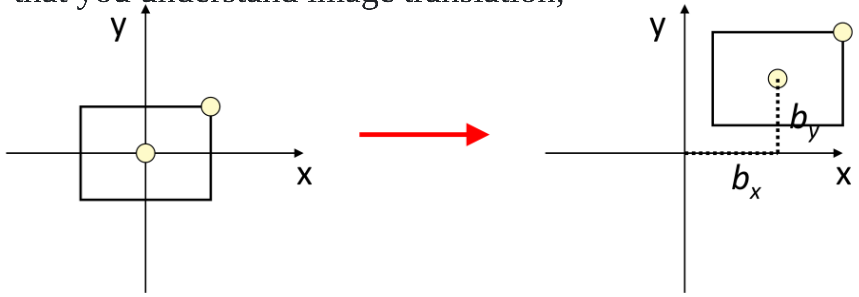
- 在下图中，漫画书与右侧图像之间的几何关系基于相似变换（旋转、平移和缩放）。
- 如果我们需要训练一个机器学习模型来找到这本漫画书，那么我们需要输入不同形状和角度的图像。
- 矩阵可以表示图像。
- 矩阵中的每个值都是特定坐标处的像素值。 可以使用矩阵乘法执行图像转换。
- 数学家已经计算出一些矩阵， 可以用来完成某些转换操作。

Translation 变换

- Image translation is the rectilinear shift of an image from a location to another, so the shifting of the of an object is called translation. The matrix shown below is used for the translation of the image:

The value of b_x defines how much the image will be moved in the x-axis and the value of b_y determines the movement of the image in the y-axis:

Now that you understand image translation,

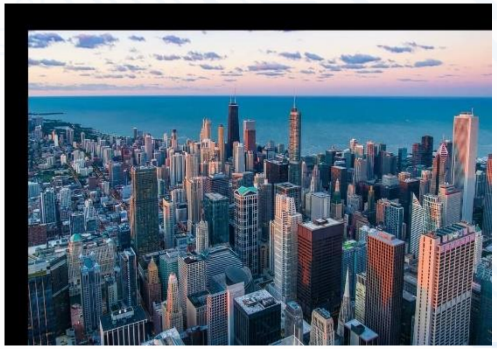


$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & B_x \\ 0 & 1 & B_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

Original image:

Translated image:

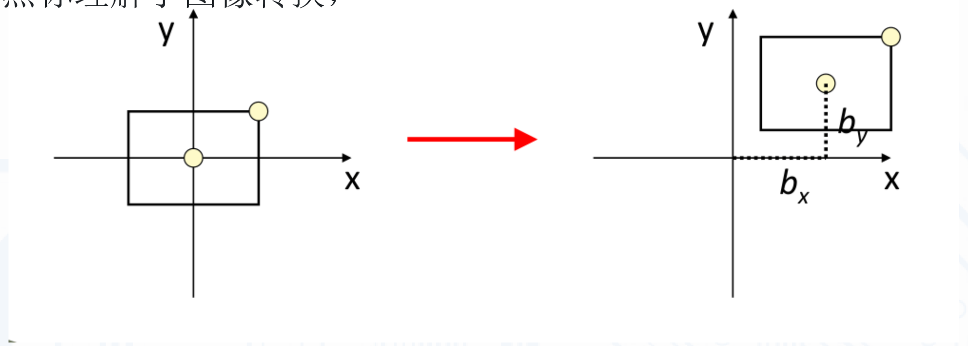


变换

- 图像平移是图像从一个位置到另一个位置的直线移动，因此物体的移动被称为平移。 下图所示的矩阵用于图像的平移：

bx的值定义了图像在x轴上的移动量，by的值决定了图像在y轴上的移动量：

既然你理解了图像转换，

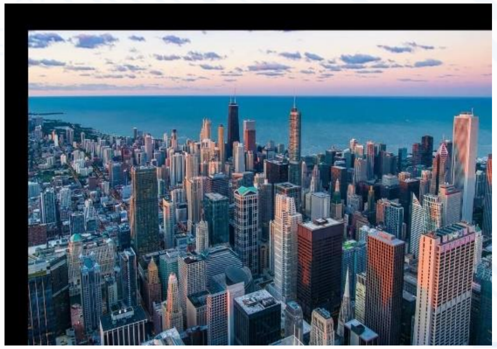


$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & B_x \\ 0 & 1 & B_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Translate

Original image:

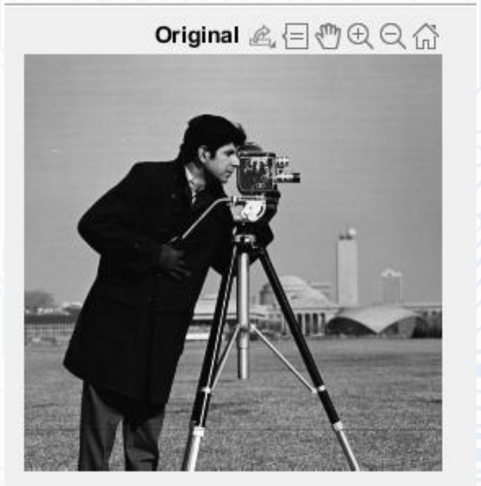
Translated image:

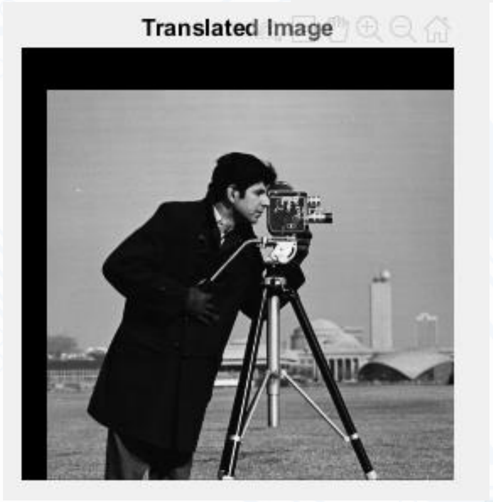


Translation

```
clc
clear all
close all
I = imread('cameraman.tif');
figure
imshow(I)
title('Original Image')
J = imtranslate(I,[15, 25]);
figure
imshow(J)
title('Translated Image')
```

Translate the image, shifting the image by 15 pixels in the *x*-direction and 25 pixels in the *y*-direction.
Note that, by default, imtranslate displays the translated image within the boundaries (or limits) of the original 256-by-256 image. This results in some of the translated image being clipped.





变换图像，在x方向上移动15像素，在y方向上移动25像素。
注意，默认情况下，imtranslate在原始256 × 256图像的边界(或限制)内显示翻译后的图像。 这导致一些翻译后的图像被剪辑。

Image Scaling 图像缩放

- Image scaling is a process used to resize a digital image.
  图像缩放是一个用来调整数字图像大小的过程。
- we will perform transformation using matrix multiplication as previously. The matrix used for scaling is shown below:
  我们将像前面一样使用矩阵乘法进行变换。 用于缩放的矩阵如下所示：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

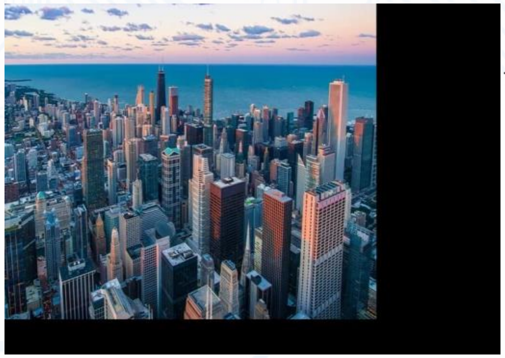
Scale

S_x and S_y are the scaling factors for x-axis and y-axis, respectively.

Sx和Sy分别是x轴和y轴的比例因子。

Original image:

Image Scaling



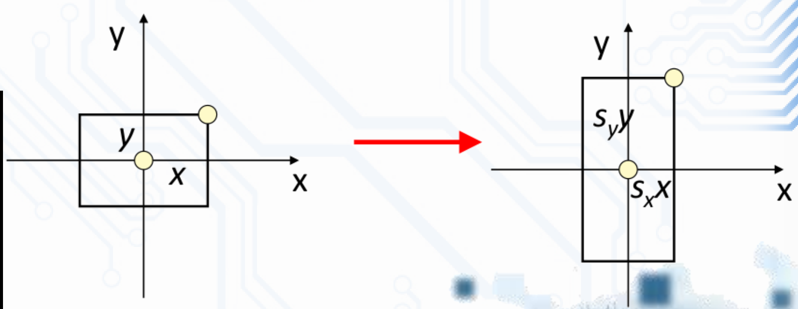


Image Scaling 图像缩放

```
clc
clear all
close all
%Load image into the workspace.
I = imread('ngc6543a.jpg');
%Shrink the image by a factor of two.
J = imresize(I, 0.5);
%Display the original image and the resized image.
figure, imshow(I), figure, imshow(J)
```

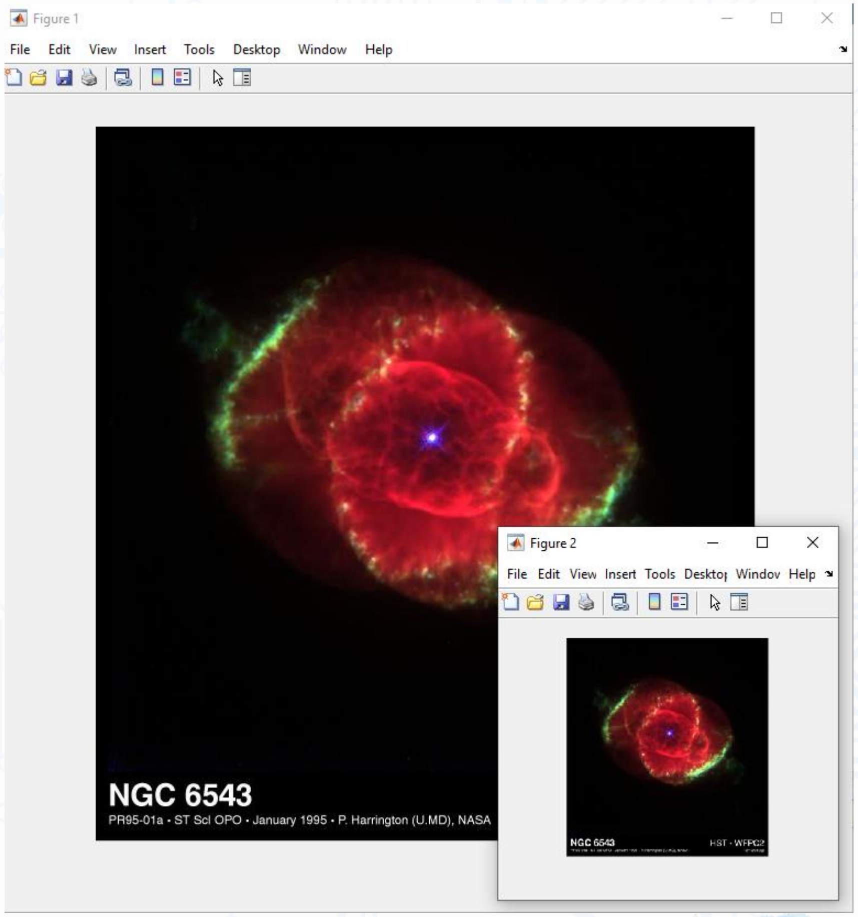


Figure 1
File Edit View Insert Tools Desktop Window Help
NGC 6543
PR95-01a · ST ScI OPO · January 1995 · P. Harrington (U.MD), NASA
Figure 2
File Edit View Insert Tools Desktop Window Help
NGC 6543
HST · WFPC2

Image Shearing

图像剪切tú xiàng jiǎn qiè

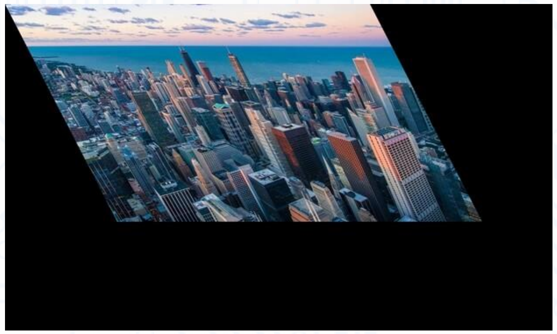
- Shear mapping is a linear map that displaces each point in fixed direction, it substitutes every point horizontally or vertically by a specific value in propotional to its x or y coordinates, there are two types of shearing effects.

剪切映射是一种线性映射，它在固定的方向上移动每个点，它在水平或垂直上用与x或y坐标成比例的特定值代替每个点，有两种类型的剪切效应。

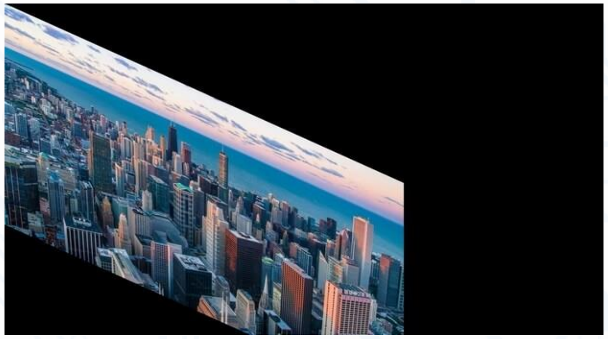
- **Shearing in the x-axis Direction x轴方向剪切**
- **Shearing in the Y-axis Direction y轴方向剪切**

Original image:

X-axis sheared image:



Y-axis sheared image:

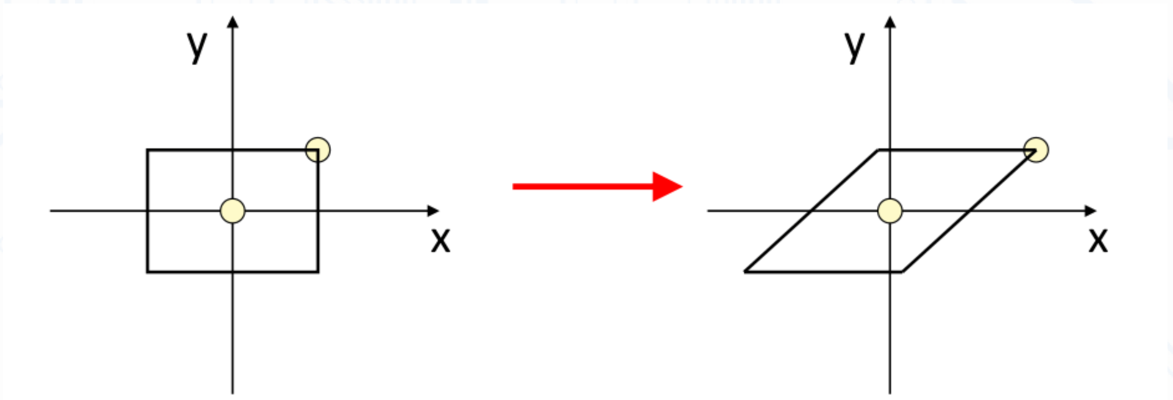


Shearing in the x-axis Direction

在 x 轴方向上剪切

- When shearing is done in the x-axis direction, the boundaries of the image that are parallel to the x-axis keep their location, and the edges parallel to y-axis changes their place depending on the shearing factor:

当在x轴方向上进行剪切时，平行于x轴的图像边界保持其位置，平行于y轴的边缘根据剪切因子的不同而改变其位置:



Shearing in the y-axis Direction

y 轴方向的剪切

- When shearing is done in the y-axis direction, the boundaries of the image that are parallel to the y-axis keep their location, and the edges parallel to x-axis changes their place depending on the shearing factor.

当在y轴方向进行剪切时，平行于y轴的图像边界保持其位置，平行于x轴的边缘根据剪切因子改变其位置。

- The matrix for shearing is shown in the below figure:

剪切矩阵如下图所示：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shear

X& Y axis sheared image:

x轴和Y轴剪切图像

```
clc
clear all
close all
I=imread('cameraman.tif');
I=double(I);

s=[2,3];
tform1 = maketform('affine',[s(1) 0 0; 0 s(2) 0; 0 0 1]);      % scaling
I1 = imtransform(I,tform1);

sh=[0.5 0.2];
tform2 = maketform('affine',[1 sh(1) 0; sh(2) 1 0; 0 0 1]);     % shear
I2 = imtransform(I,tform2);

theta=3*pi/4;                                % rotation
A=[cos(theta) sin(theta) 0; -sin(theta) cos(theta) 0; 0 0 1];
tform3 = maketform('affine',A);
I3 = imtransform(I,tform3);

figure
subplot(2,2,1),imagesc(I),axis image
title('Original','FontSize',18)
subplot(2,2,2),imagesc(I1),axis image
title('Scaled','FontSize',18)
subplot(2,2,3),imagesc(I2),axis image
title('Shear','FontSize',18)
subplot(2,2,4),imagesc(I3),axis image
title('Rotation','FontSize',18)
colormap(gray)
```

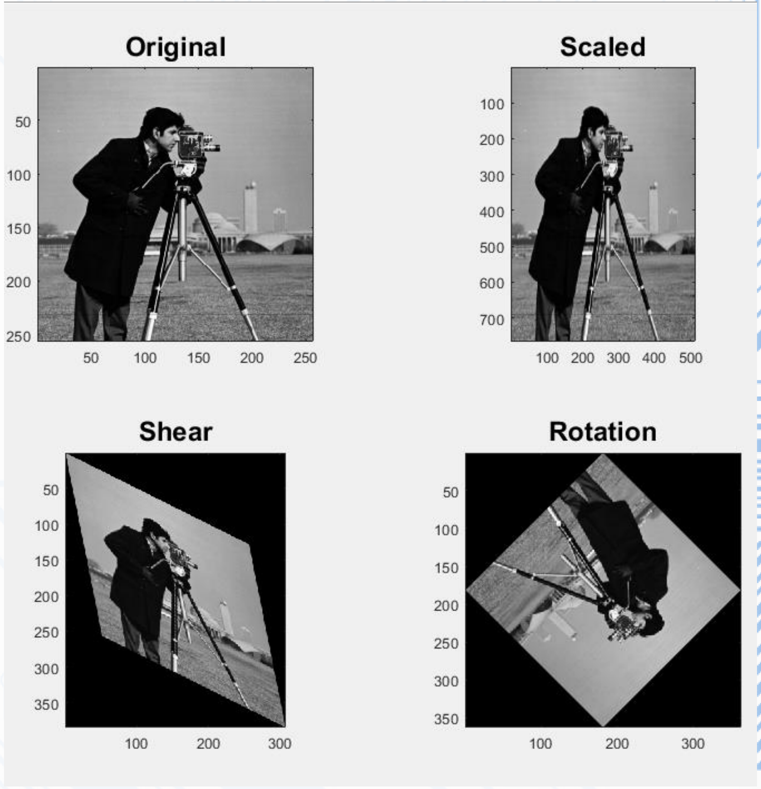


Image Reflection

图像反射tú xiàng fǎn shè

- Image reflection (or mirroring) is useful for flipping an image, it can flip the image vertically as well as horizontally, it is a particular case of scaling. For reflection along the x-axis, we set the value of Sy to -1,

  and Sx to 1 and vice-versa for the y-axis reflection.

  图像反射(或镜像)对翻转图像很有用，它可以垂直翻转图像，也可以水平翻转图像，这是缩放的一种特殊情况。 对于沿x轴的反射，我们将Sy的值设为-1,Sx设为1，对于y轴的反射，反之亦然。

- The transformation matrix for reflection is shown below:

  反射变换矩阵如下:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & \text{rows} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Reflection x-axis

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & \text{cols} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Reflection y-axis

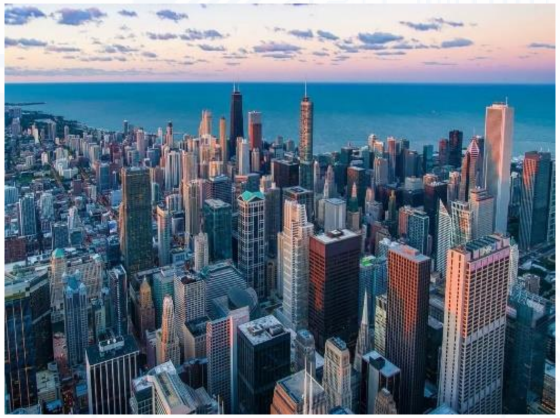
Image Reflection

图像反射

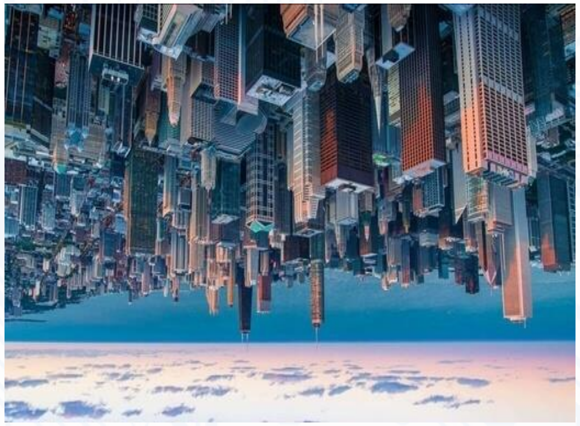
As previously, this will reflect it x-axis, if you want y-axis reflection, uncomment the second matrix and comment the first one.

如前所述，这将反映它的x轴，如果你想要y轴反射，取消注释第二个矩阵，并注释第一个。

Original image:



X-axis reflected image:



Y-axis reflected image:

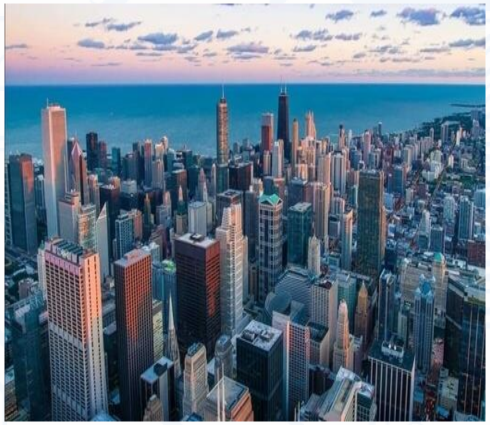


Image Reflection

图像反射

```
clc
clear all
close all
 I = imread('onion.png');
I2 = I(:,end:-1:1,:);          %# horizontal flip
I3 = I(end:-1:1,:,:);          %# vertical flip
I4 = I(end:-1:1,end:-1:1,:);    %# horizontal+vertical flip

subplot(2,2,1), imshow(I)
subplot(2,2,2), imshow(I2)
subplot(2,2,3), imshow(I3)
subplot(2,2,4), imshow(I4)
```

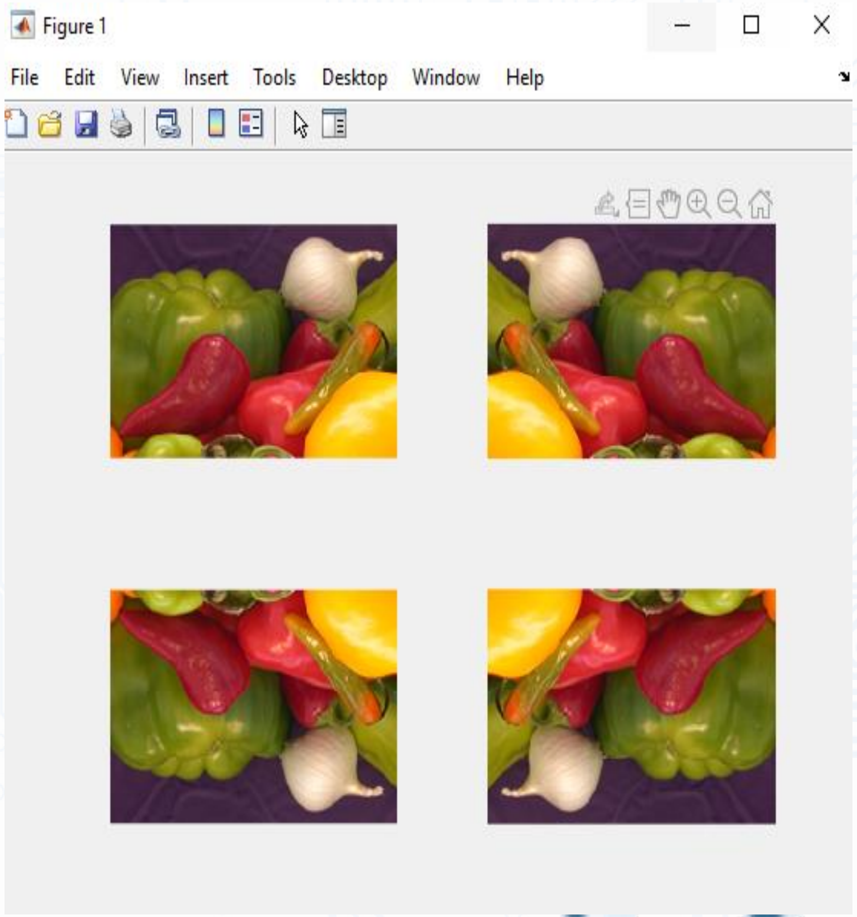


Image Rotation

图像旋转tú xiàng xuán zhuǎn

- Rotation is a concept in mathematics that is a motion of a certain space that preserves at least one point. Image rotation is a common image processing routine with applications in matching, alignment, and other image-based algorithms, it is also extensively in data augmentation, especially when it comes to image classification.

旋转是数学中的一个概念，它是在一定的空间中保持至少一个点的运动。图像旋转是一种常见的图像处理程序，应用于匹配、对齐和其他基于图像的算法，在数据增强中也有广泛的应用，特别是在图像分类中。

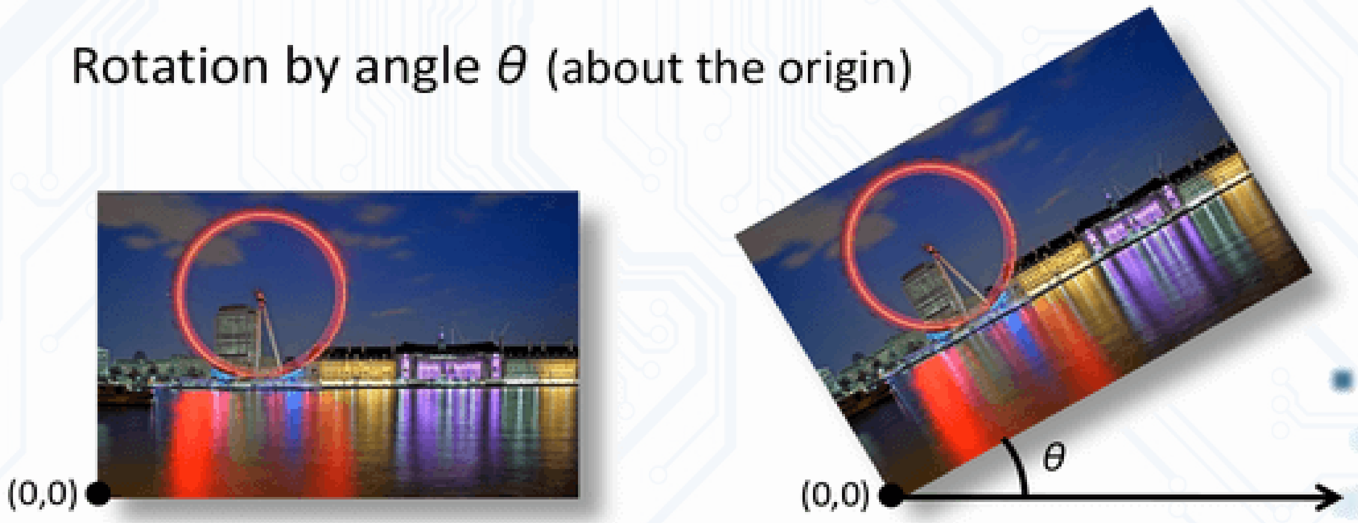


Image Rotation

图像旋转

- The transformation matrix of rotation is shown in the below figure, where theta (θ) is the angle of rotation:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2D *in-plane* rotation

旋转变换矩阵如下图所示，其中θ (θ)为旋转角度:

Original image:

Output image:

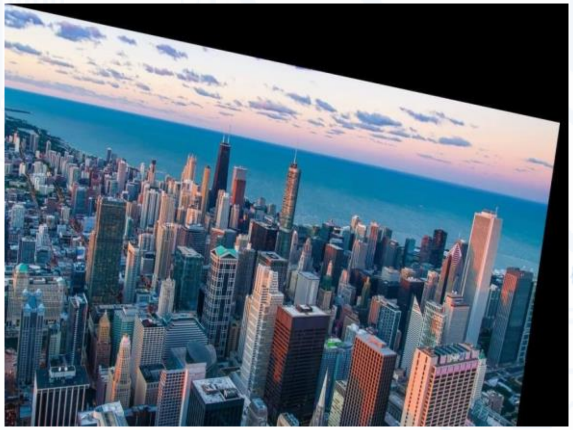


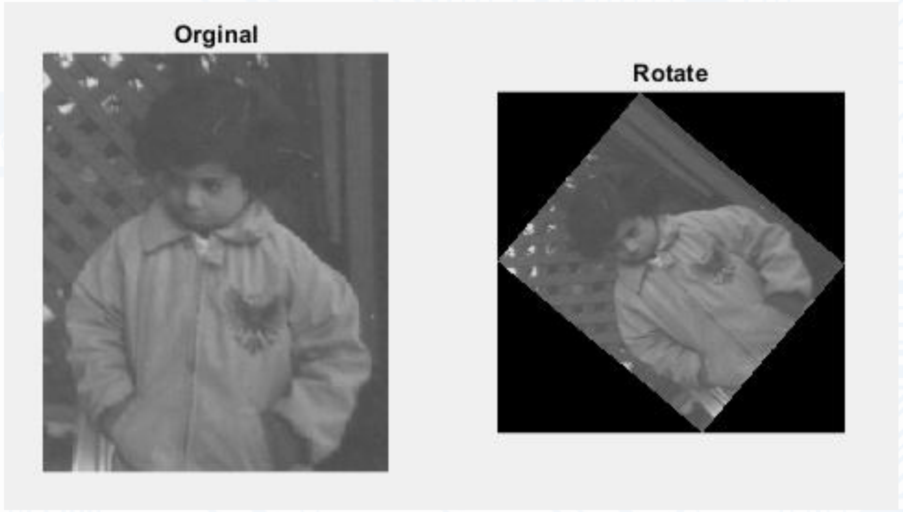
Image Rotation 图像旋转

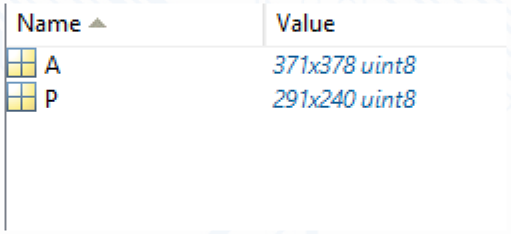
```
clc
clear all

P =imread('pout.tif');
A=imrotate(P,50);

subplot(1,2,1)
 imshow(P)
title('Orginal')

subplot(1,2,2)
 imshow(A)
title('Rotate')
```





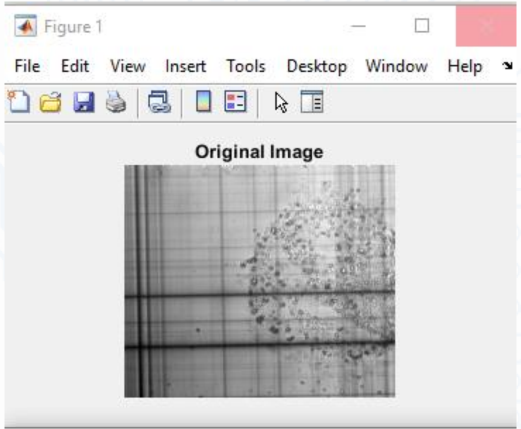
```
Name ▲      Value
A           371x378 uint8
P           291x240 uint8
```

Size of image after rotation changed
旋转后的图像大小改变

Image Rotation

图像旋转

```
clc
clear all
close all
 %Read an image into the workspace, and convert it to a grayscale image.
%将图像读入工作空间，并将其转换为灰度图像。
I = fitsread('solarspectra.fts');
I = rescale(I);
 %Display the original image.
figure
imshow(I)
title('Original Image')
 %Rotate the image 10 degree clockwise to bring it into better horizontal
alignment. The example specified bilinear interpolation and requests that the
result be cropped to be the same size as the original image.
%顺时针旋转图像10度，使其进入更好的水平对齐。  该示例指定了双线性
插值，并要求将结果裁剪成与原始图像相同的大小。
J = imrotate(I,-10,'bilinear','crop');
 %Display the rotated image.
figure
imshow(J)
title('Rotated Image')
```



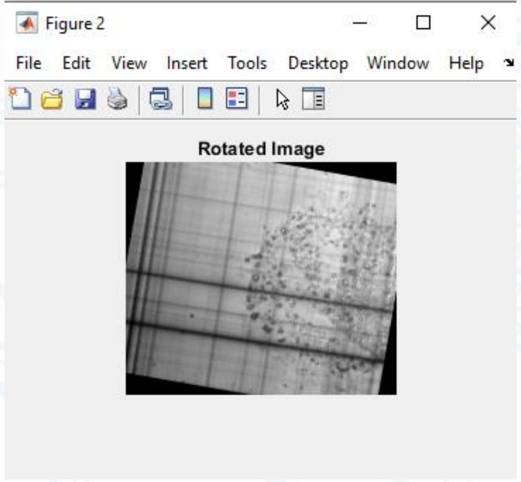


Image Cropping

图像裁剪tú xiàng cái jiǎn

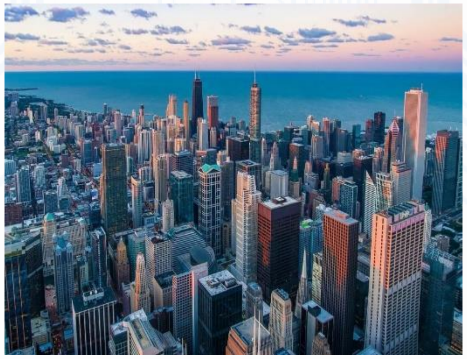
- Image cropping is the removal of unwanted outer areas from an image, a lot of the above examples introducted black pixels, you can easily remove them using cropping. The below code does that:
- we can crop the image simply by indexing the array, in our case, we chose to get 200 pixels from 100 to 300 on both axes, here is the output image:

Original image:

图像裁剪

- 图像裁剪是去除图像不需要的外部区域，上面的很多例子都引入了黑色像素，你可以很容易地使用裁剪去除它们。下面的代码做到了这一点:
- 我们可以简单地通过索引数组来裁剪图像，在我们的例子中，我们选择在两个轴上从100到300得到200像素，这是输出图像:

Original image:



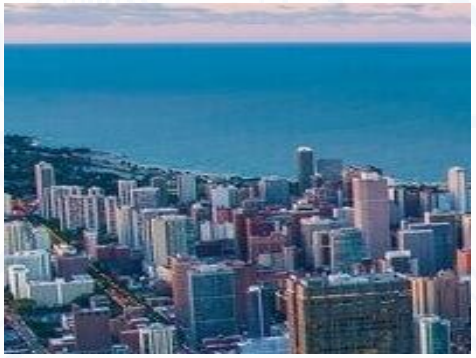
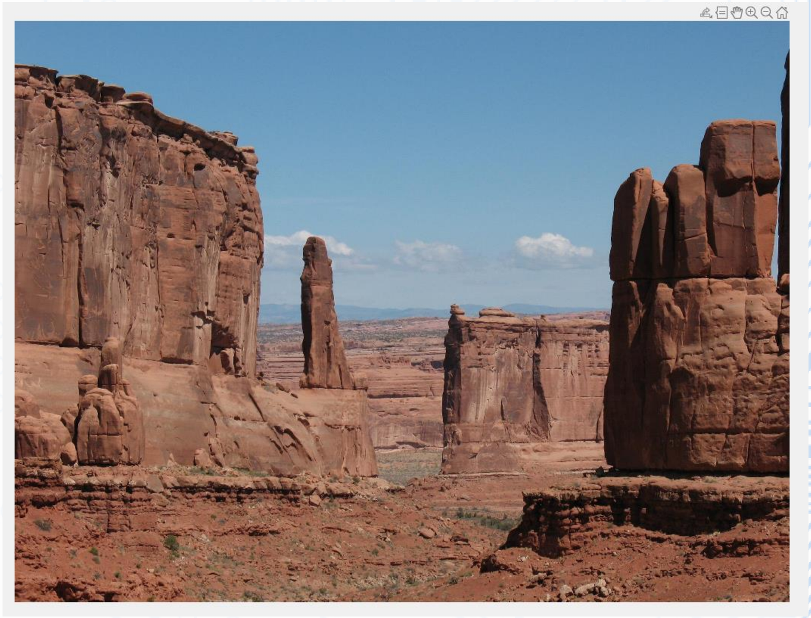


Image Cropping

图像裁剪

```
 clc
clear all
close all
% Read and display an image.
I = imread('parkavenue.jpg');
figure(1);imshow(I)

 J = imcrop(I,[20 200 800 700]);
figure(2);imshow(J)
```



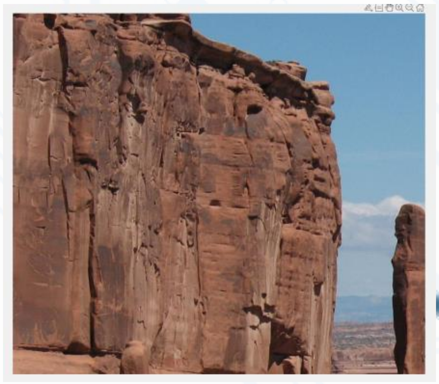


Image Cropping

图像裁剪

```
clc
clear all
close all
% Read and display an image.
I = imread('parkavenue.jpg');
figure(1);imshow(I)
%Specify a target window size as a two-element vector of the form
 [width, height].
%指定窗体的两个元素向量作为目标窗口大小(宽度、高度)。
targetSize = [300 600];
%Create a Rectangle object that specifies the spatial extent of the crop
window.
%创建一个矩形对象，指定裁剪窗口的空间范围。
r = centerCropWindow2d(size(I),targetSize);
%Crop the image to the spatial extents. Display the cropped region.
%裁剪图像到空间范围。显示裁剪区域。
J = imcrop(I,r);
figure(2);imshow(J)
```

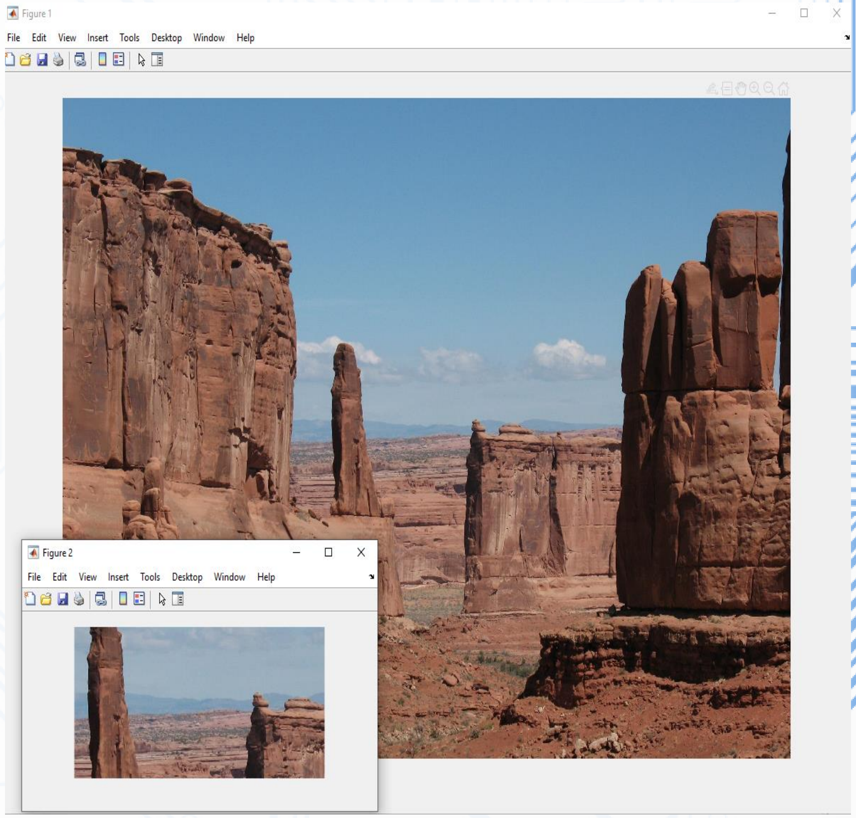


Figure 1
File Edit View Insert Tools Desktop Window Help
Figure 2
File Edit View Insert Tools Desktop Window Help

MAY-BRITT MOSER
Nobel Prize in Physiology or Medicine 2014

Private/Kavli Institute for Systems Neuroscience

“I learned at an early age that work makes you happy.”

Student Task_3: DIP

- 请帮我翻译部分的朋友鼓掌
- Qǐng bāng wǒ fānyì bùfèn de péngyǒu gǔzhǎng

Solve the Question shared in mooc

解决mooc分享的问题

Send for Next lecture

发送下一个讲座

Reference 参考

Introduction to MATLAB, *Kadin Tseng,Boston University, Scientific Computing and Visualization*

MATLAB入门，Kadin Tseng，波士顿大学，科学计算与可视化

- Images taken from Gonzalez & Woods, Digital Image Processing (2002)

从Gonzalez & Woods拍摄的图像，数字图像处理(2002)

ages taken from Gonzalez & Woods, Digital Image Processing (2002)

江西理工大学

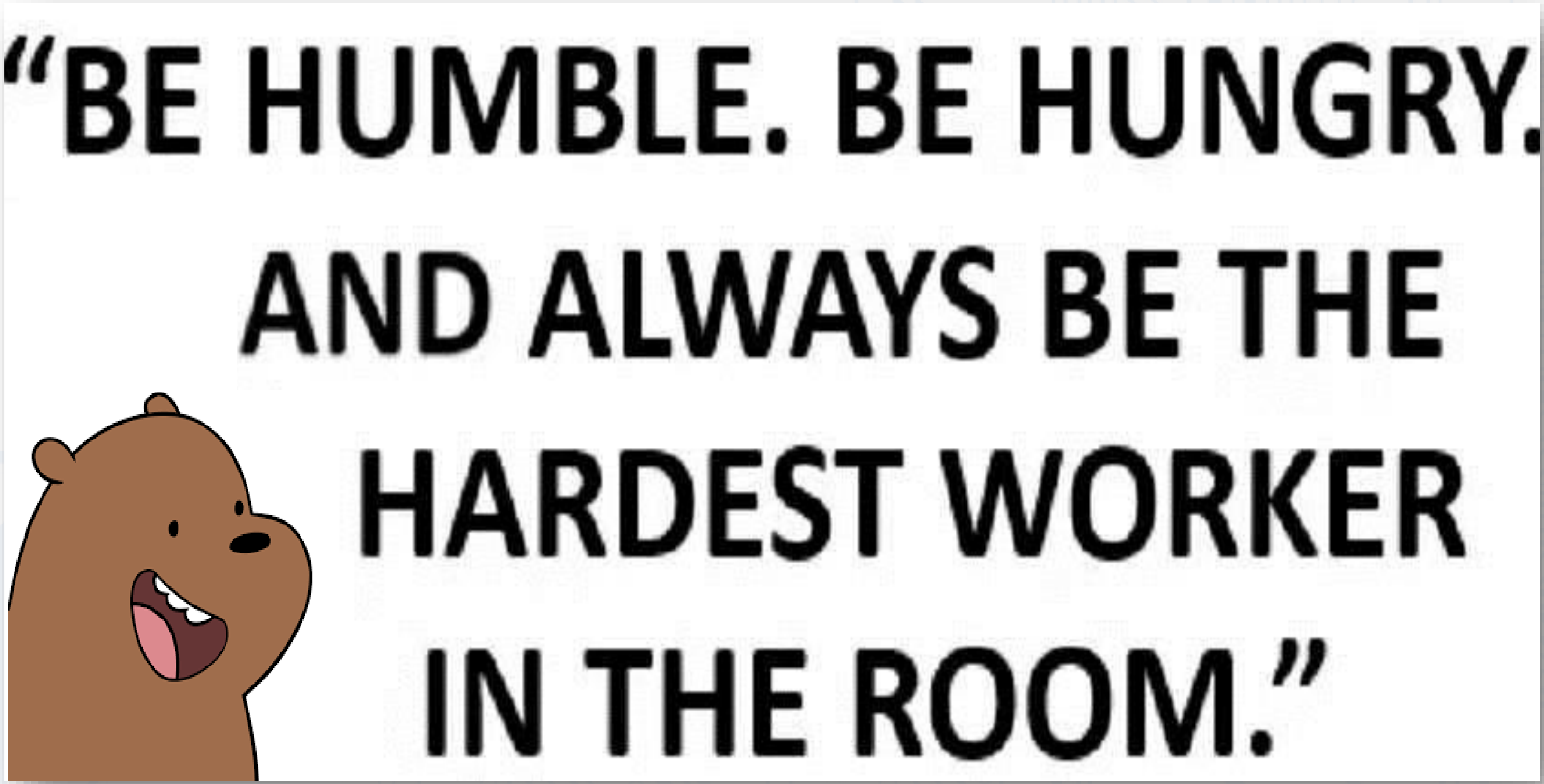
Jiangxi University of Science and Technology

信息工程学院

School of information engineering

Digital Image Processing

THANK YOU


"BE HUMBLE. BE HUNGRY.
AND ALWAYS BE THE
HARDEST WORKER
IN THE ROOM."