

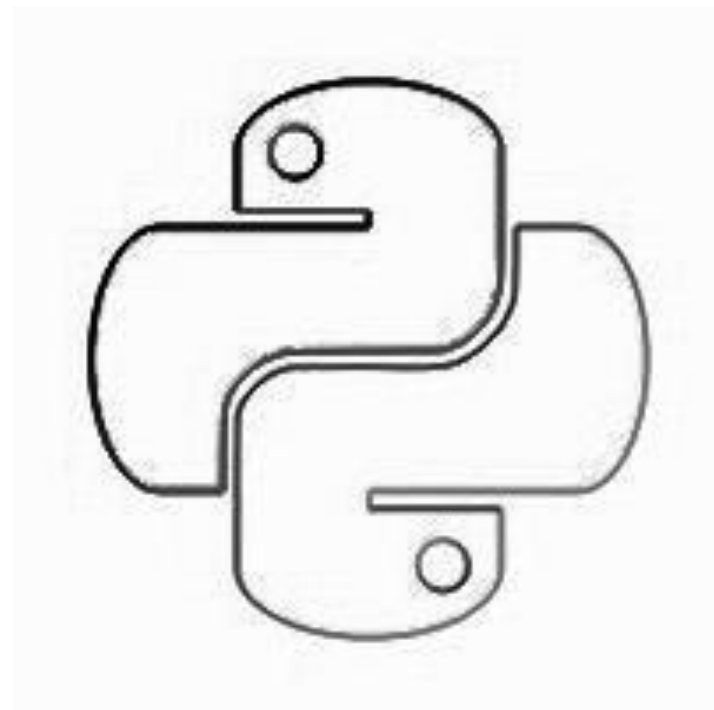


“图像的手绘效果”实例分析

“图像的手绘效果”实例介绍

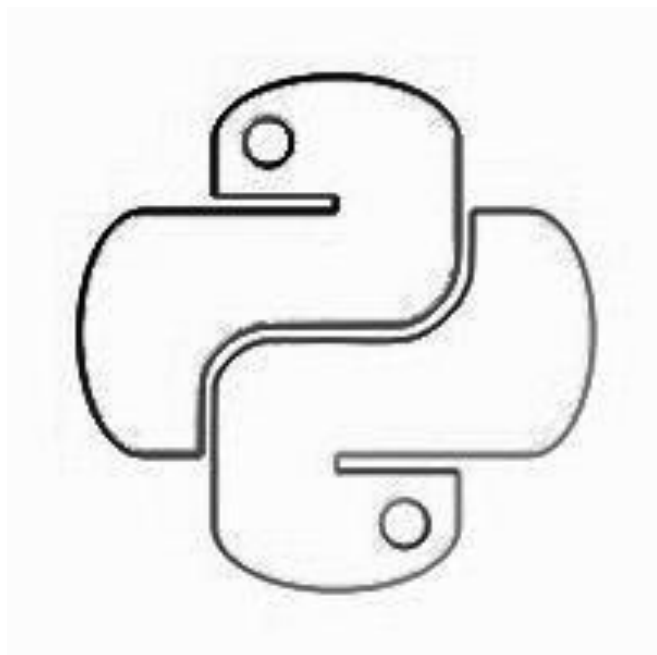


原图



手绘效果图

“图像的手绘效果” 实例分析



手绘效果的几个特征：

- 黑白灰色
- 边界线条较重
- 相同或相近色彩趋于白色
- 略有光源效果

“图像的手绘效果” 代码

```
from PIL import Image
import numpy as np

a = np.asarray(Image.open('D:/pycodes/beijing.jpg').convert('L')).astype('float')

depth = 10.                                # (0-100)
grad = np.gradient(a)                       # 取图像灰度的梯度值
grad_x, grad_y = grad                      # 分别取横纵图像梯度值
grad_x = grad_x*depth/100.
grad_y = grad_y*depth/100.
A = np.sqrt(grad_x**2 + grad_y**2 + 1.)
uni_x = grad_x/A
uni_y = grad_y/A
uni_z = 1./A

vec_el = np.pi/2.2                         # 光源的俯视角度，弧度值
vec_az = np.pi/4.                          # 光源的方位角度，弧度值
dx = np.cos(vec_el)*np.cos(vec_az)         # 光源对x 轴的影响
dy = np.cos(vec_el)*np.sin(vec_az)         # 光源对y 轴的影响
dz = np.sin(vec_el)                        # 光源对z 轴的影响

b = 255*(dx*uni_x + dy*uni_y + dz*uni_z)    # 光源归一化
b = b.clip(0,255)

im = Image.fromarray(b.astype('uint8'))     # 重构图像
im.save('D:/pycodes/beijingHD.jpg')
```

“图像的手绘效果”实例编写



梯度的重构

利用像素之间的梯度值和虚拟深度值对图像进行重构

根据灰度变化来模拟人类视觉的远近程度

```
depth = 10.  
grad = np.gradient(a)  
grad_x, grad_y = grad  
grad_x = grad_x*depth/100.  
grad_y = grad_y*depth/100.
```



预设深度值为10 取值范围0-100



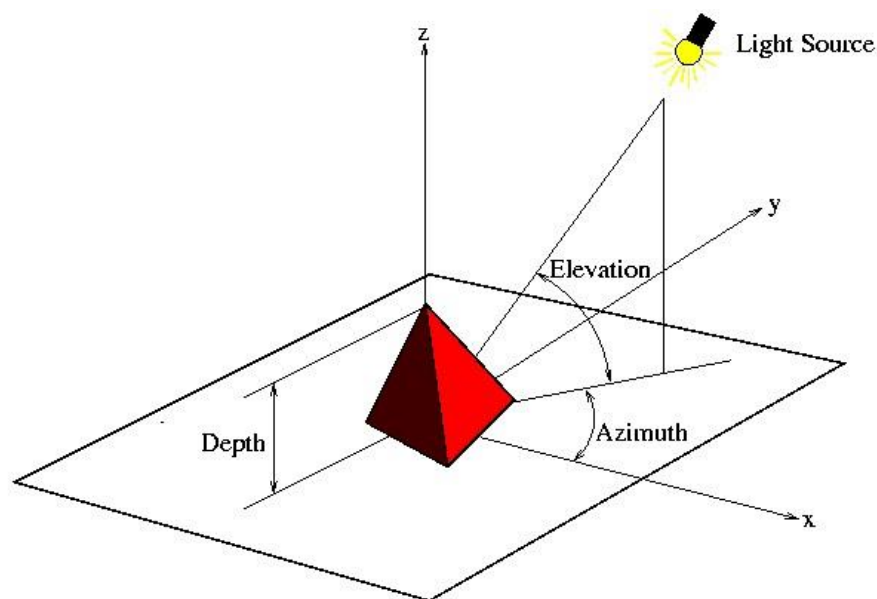
提取x和y方向的梯度值



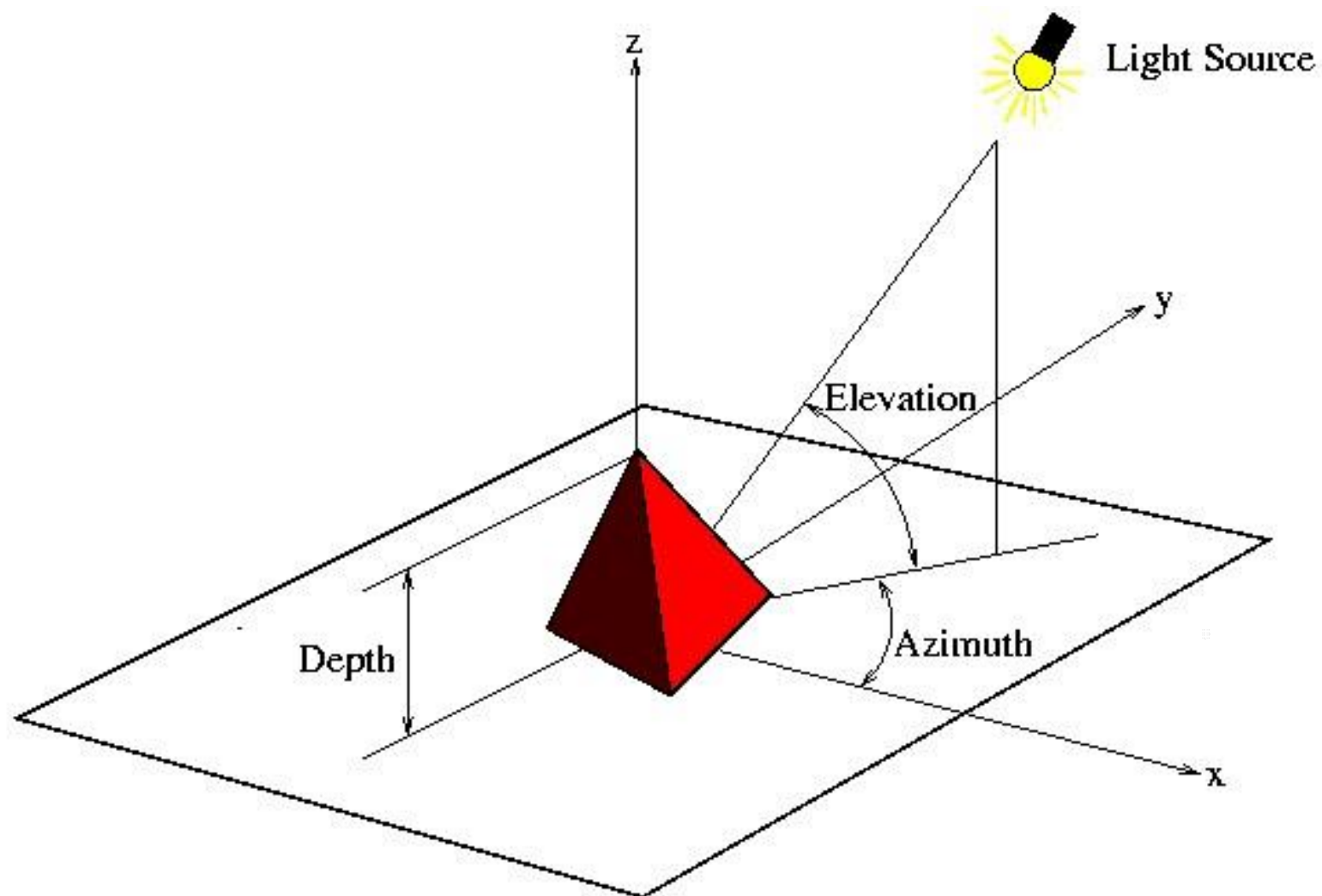
根据深度调整x和y方向的梯度值

光源效果

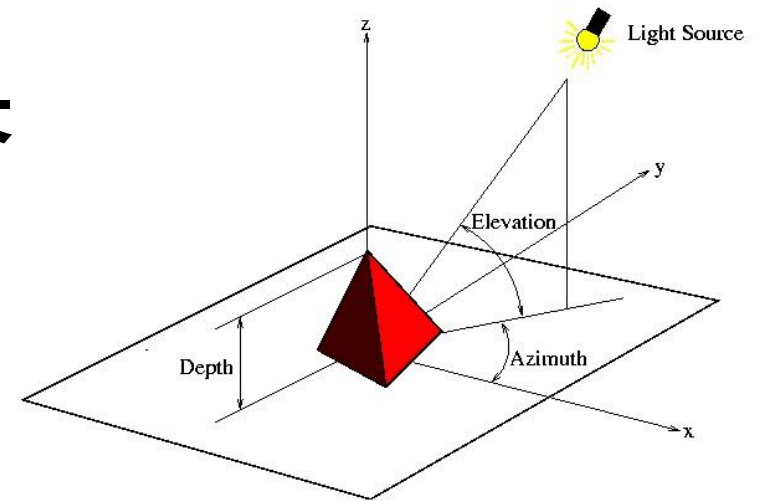
根据灰度变化来模拟人类视觉的远近程度



- 设计一个位于图像斜上方的虚拟光源
- 光源相对于图像的俯视角为Elevation , 方位角为Azimuth
- 建立光源对个点梯度值的影响函数
- 运算出各点的新像素值



光源效果



```
vec_el = np.pi/2.2  
vec_az = np.pi/4.  
dx = np.cos(vec_el)*np.cos(vec_az)  
dy = np.cos(vec_el)*np.sin(vec_az)  
dz = np.sin(vec_el)
```

$\text{np.cos}(\text{vec_el})$ 为单位光线在地平面上的投影长度

dx , dy , dz 是光源对 $x/y/z$ 三方向的影响程度

梯度归一化

构造x和y轴梯度的三维归一化单位坐标系

```
A = np.sqrt(grad_x**2 + grad_y**2 + 1.)  
uni_x = grad_x/A  
uni_y = grad_y/A  
uni_z = 1./A  
b = 255*(dx*uni_x + dy*uni_y + dz*uni_z)
```

梯度与光源相互作用，将梯度转化为灰度

图像生成

为避免数据越界，将生成的灰度值裁剪至0-255区间



```
b = b.clip(0,255)
im = Image.fromarray(b.astype('uint8'))
im.save('./beijingHD.jpg')
```



生成图像