水色图

PhotoShop方法

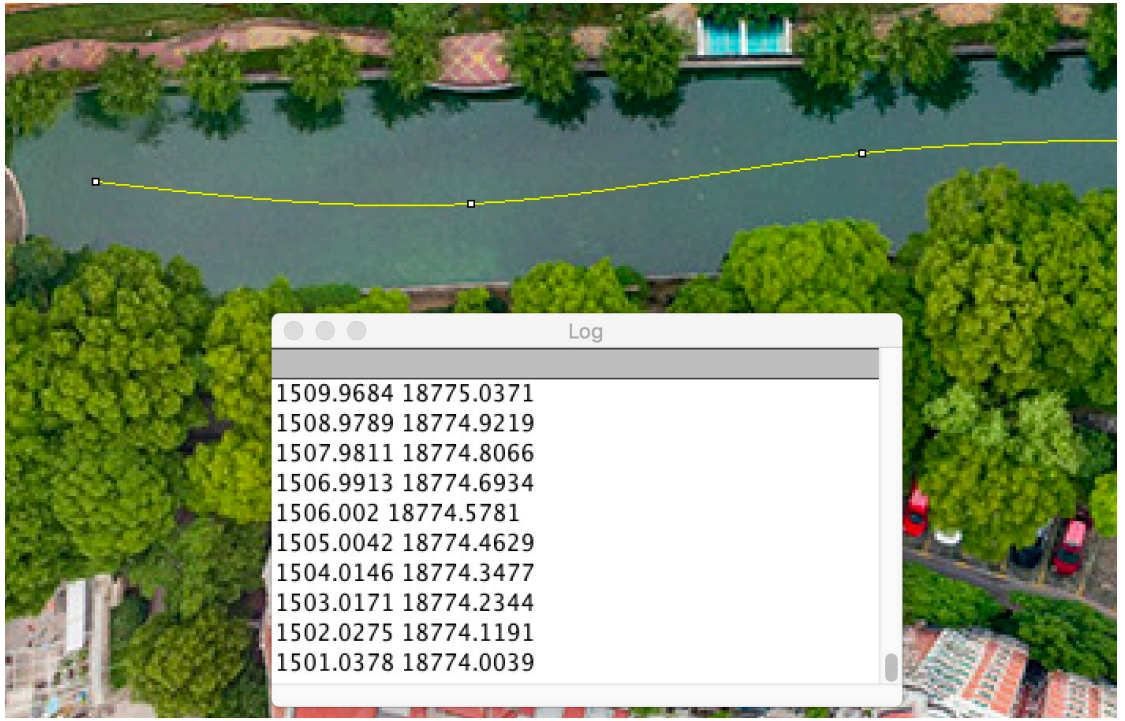
- 拉直：分别将河道中水平段截取出来，转弯处的拉直后，依次拼在一起。工作量大。
- 等间隔取色：每隔固定距离后，选取该位置周围固定范围的平均颜色，合成新的水色图。不方便计算等间隔长度，转弯处的长度。

优化方案

通过手动在河道中画线，选择需要的河道区域，能够读出经过的每点坐标位置

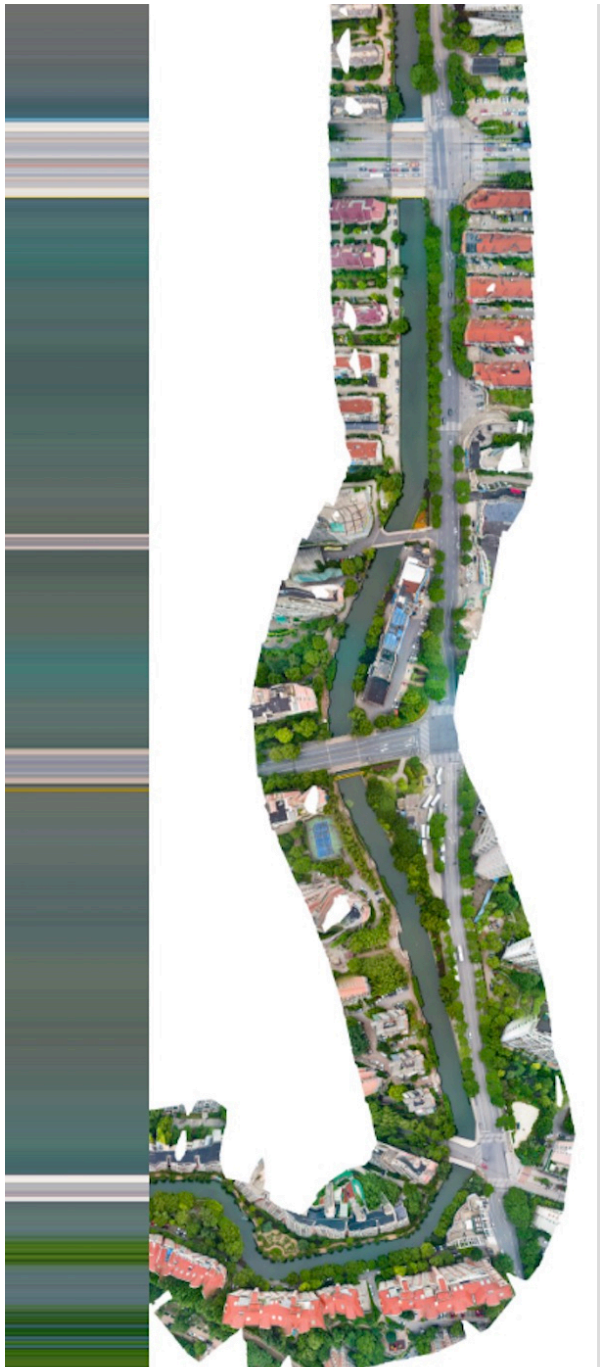
Macro.ijm.ijm.ijm

```
run("Fit Spline", "straighten");
getSelectionCoordinates(x, y);
values = getProfile();

for (i=0; i<values.length; i++)
  print(x[i],y[i]);
```



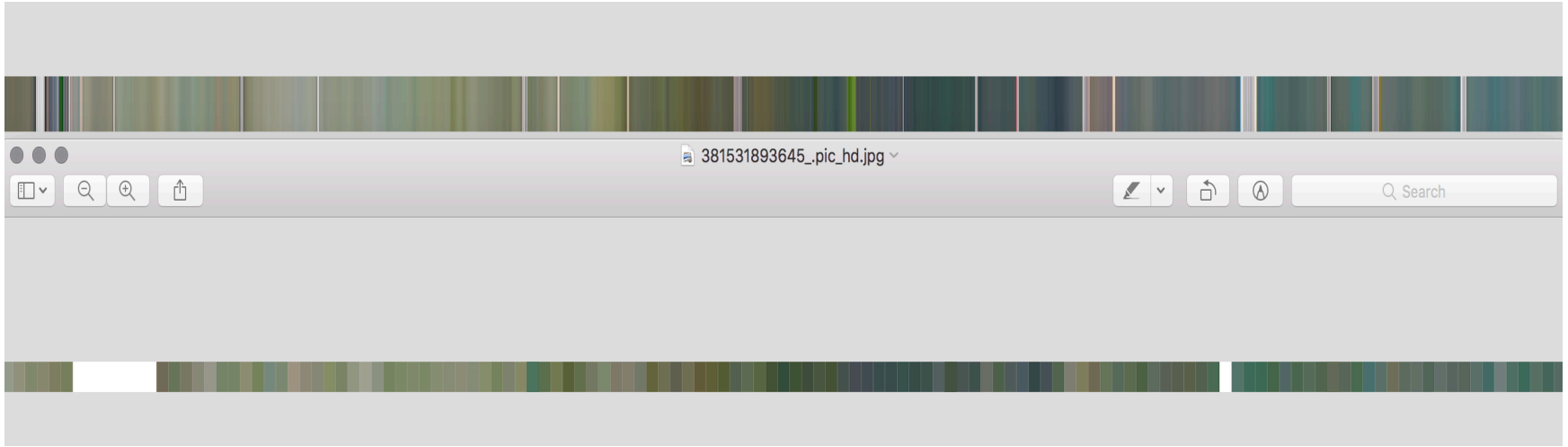
```
def readtext(filename):
    # read the position file
    f= open(filename, 'r')
    Line=f.readlines()
    x = []
    y = []
    for line in Line:
        line = line.split(' ')            # seperate the list with white space
        x.append(line[0])                 # x coordinate
        y.append(line[1])                 # y coordinate

    # turn the input from string to int
    X = []
    Y = []
    for i in range(len(x)):
        X_temp = round(float(x[i]))
        X.append(X_temp)
        Y_temp = round(float(y[i]))
        Y.append(Y_temp)
    return X,Y
```

```
def drawImage(image,X,Y,length):
    a = length
    pix = im.load()                    # read the image
    RGB = []
    for i in range(len(X)):
        RGB.append(pix[X[i],Y[i]])     # Get the RGB Value of the a pixel of an image

    size = (len(Y),a)                  # set the new picture size
    im2 = Image.new(im.mode, size)     # create a new image to show your color
    draw = ImageDraw.Draw(im2)
    for i in range(len(X)):            #  draw the line from the first pixel, each color is 1 pixel in width
        draw.line([(i, 0),(i,a)] , fill=RGB[i], width = 1)
    im2.show()
    im2.save('color.JPG')              # file name of the output image
                                       # file path is the same as this program


#%%
filename = "/Users/tian/Desktop/untitled folder/Log.txt"  # Name and path of the position file
im = Image.open('/Users/tian/Desktop/untitled folder/20180529-小吉浦(全段)_90m.jpg') # Name and path of the image
a = 500                                # length of each line
X,Y = readtext(filename)
drawImage(im,X,Y,a)
```



可以根据位置坐标，读出该点的颜色，通过这每一个像素点的颜色合成新的水色图

输出结果



上：程序生成的水色图

下：等间隔取色制作的

河道长度误差

- 小吉浦总长6.39千米，按照平行于河岸所画线与按需求画线的长度比较，得出长度误差在2%范围内，即133米内

平均颜色误差

上：取单点颜色

下：周围附近像素的平均颜色

```
        rgb = sRGBColor(a,b,c)
        lab = convert_color(rgb, LabColor,through_rgb_type=AdobeRGBColor)
        Lab.append(lab)

        RGB_original = pix[X[i],Y[i]]
        A = int(RGB_original[0])
        B = int(RGB_original[1])
        C = int(RGB_original[2])
        rgb_ori = sRGBColor(A,B,C)
        lab_ori = convert_color(rgb_ori, LabColor,through_rgb_type=AdobeRGBColor)
        Lab_ori.append(lab_ori)

        delta_e = delta_e_cie1976(lab, lab_ori)
        error.append(delta_e/100)

    print( np.mean(error) )
    print ( np.std(error))
```

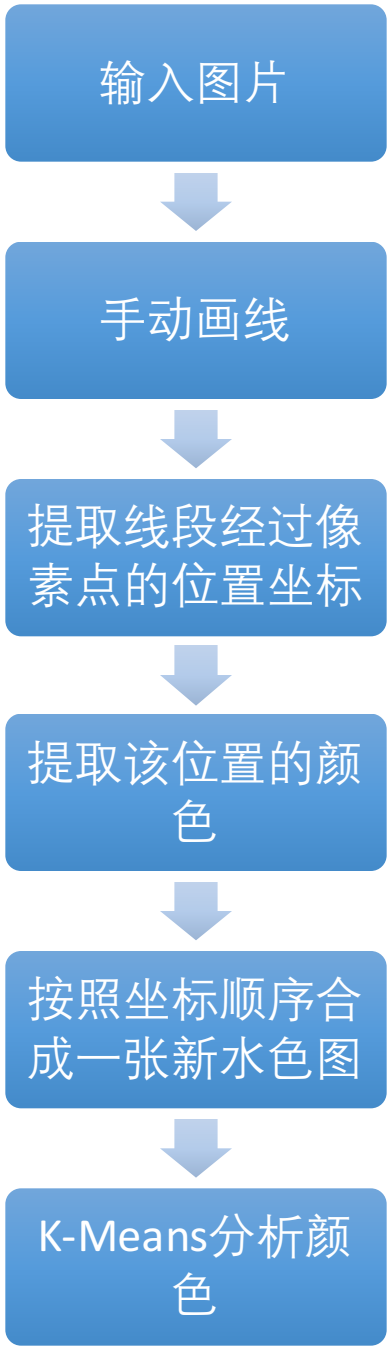
- ΔE是衡量在均匀颜色感觉空间中，人眼感觉色差的测试单位，能够精确将色彩还原的准确性量化为一个数值。ΔE在小于1.0时，属于人眼无法观察到的色差。当ΔE在1～2之间时，色差需要密切观察才能够感知到。因此颜色误差几乎不影响水色图结果。
- ΔE平均值在1.5～2的范围内，标准差在5之内。

输出结果分析

提取后的水色图，可以观察到整体颜色变化趋势。K-Means对于大量的未知标注数量集，计算颜色间的相似程度。通过这种方法能够更加直观的看出水色变化趋势以及河道该段的平均颜色。

```
print(__doc__)
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.utils import shuffle
from PIL import Image

n_colors = 5
# Load the photo
im = Image.open('/Users/tian/Desktop/untitled folder/高阶水色拉伸/20180531-小吉浦(全段)_90mcolor.jpg')
im = np.array(im, dtype=np.float64) / 255

# Load Image and transform to a 2D numpy array.
w, h, d = original_shape = tuple(im.shape)
assert d == 3
image_array = np.reshape(im, (w * h, d))
# Set training set as random 10000
image_array_sample = shuffle(image_array, random_state=0)[:10000]
kmeans = KMeans(n_clusters=n_colors, random_state=0).fit(image_array_sample)
```

```
# Get labels for all points
labels_500 = kmeans.predict(image_array)
def recreate_image(codebook, labels, w, h):
    """Recreate the (compressed) image from the code book & labels"""
    d = codebook.shape[1]
    image = np.zeros((w, h, d))
    label_idx = 0
    for i in range(w):
        for j in range(h):
            image[i][j] = codebook[labels[label_idx]]
            label_idx += 1
    return image

plt.figure()
plt.clf()
ax = plt.axes([0, 0, 1, 1])
plt.axis('off')
plt.imshow(recreate_image(kmeans.cluster_centers_, labels_500, w, h))
plt.savefig('color_kmeans.png')
```


输入图片
手动画线
提取线段经过像素点的位置坐标
提取该位置的颜色
按照坐标顺序合成一张新水色图
K-Means分析颜色

流程如图所示