EEE-6512 Image Processing and Computer Version

Homework #3

Wenxuan Wang

UF gatorlink username: wenxuanwang

Ufmail: wenxuanwang@ufl.edu

UFID:64118211

**Part1 Textbook Qustions**

一些文字和图片的手机截图

描述已自动生成

手机屏幕截图

描述已自动生成

图片包含 表格

描述已自动生成

手机屏幕截图

描述已自动生成

图示

描述已自动生成

**Part 2: code**

Explanation of my code

文本

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

Firstly, we should read the video and the parameter. And then initialize the paramteter and set the threshould value. I set the threshould value to 100. （This threshould value was determined after I tried many different values. I think that the value from 80-100 performs well.）Then it is a for loop, that we should firstly read the first 100 frames. And use double to read the video frams as images. And split three channels from the rgb images. Then I use the 1/3(r + g + b) [1]to convert rgb to grayscale. I used this conversion method because it is simple to compute and convenient. Besides, through this method, this kind of picture is closer to the real state of the object under the light. Although 1/4(r + 2g + b)is more suitable for human eyes to distinguish, but because the proportion of green increases, it may cause the image color difference to be too large, which is not conducive to subsequent image processing。 And in this loop, I compute the sum of the computed grayscale images. And use uint8 to get integer and plot the average image. Then it is the second for loop. I load the 1st, 20th, 40th, 80th, 100th frame of the video as rgb image. And all the same, I split the channels, and convert rgb to get integer. Then I compute the image after background substraction. I set the intensity which is greater than or equal to the threshould 255 and set the intensity which is lower than then threshould 0. Finally, I plot the image after background substraction. I also add the original one to compare.

**Result image:**

**Average image:**

模糊的黑白照片

描述已自动生成

**The five thresholded background subtraction results images:**

图片包含 图形用户界面

描述已自动生成

图片包含 图形用户界面

描述已自动生成

图片包含 照片, 游戏机, 人们, 男人

描述已自动生成

社交网络的手机截图

描述已自动生成

社交网络的手机截图

描述已自动生成

Explain your results. Where does your algorithm perform well and where does it perform poorly? Why?

I think the algorithm performs poorly because it works bad for converting the first 100 frames of the video. We aim to clear the background through averaging the image, but in the average image, the shadow of people still exists. I think if we use a full-long video or make the frames as more as possible, the performance may be better. Besides, while writing the code, we can modify suitable threshould to let the image perform well.

Reference

[1] B. H. Brown, R. H. Smallwood, D. C. Barber, P. V Lawford, and D. R. Hose, *Image processing and analysis*. 2004.