Project 0: Using Python for Image Manipulation

Author: Gabriel Hofer

CSC-414 Introduction to Computer Vision

Instructor: Dr. Hoover

Due: January 27, 2020

Department: Computer Science and Engineering
University: South Dakota School of Mines and Technology

1 Questions

1. We wish to set all pixels that have a value of 10 or less to 0, to remove camera sensor noise. However, our code is slow when run on a database with 1000 grayscale images.

(a) How could we speed it up? Please include your code.

From the CSC-414 Python Tutorial: "Since NumPy is an extension for Python that is written in C, NumPy operations are faster than their corresponding Python equivalents. For example, performing matrix multiplication of two NumPy arrays is faster than iterating through two Python lists representing arrays and multiplying the correct elements. As such, we recommend doing as much of your calculations in NumPy as possible. It is best to avoid using for loops whenever possible; one can attain significant performance improvements through vectorization and logical indexing."

Listing 1: Logical Indexing of Greyscale Image

```
# start the timer
start = time.time()

def process(file):
  A = io.imread(file)
  (m1,n1) = A.shape

  plt.imshow(A)
  plt.show()

  # logical indexing to set values less than 10 to 0
  B = A < 80
  A[B] = 0

  # display the image after changes
  plt.imshow(A)
  plt.show()

  # stop the timer and display elapse time to modify image
  end = time.time()
  print("total time: "+str(end - start))

# read the image
for file in sys.argv[1:]:
    process(file)
```

(b) What factor speedup would we receive over 1000 images? Please measure it and include your code

(c) Next, we wish to operate on color images. How does your speeded-up version from 1 (a) change for color images? Please implement and measure it, report the speed factor change, and include your code.

2. Suppose we wish to reduce the brightness of an image by editing the values in its matrix. But, when trying to visualize the result, we see some errors.

(a) What is incorrect with this approach? How can it be fixed while maintaining the same intended brightness reduction? Please include your code and result image.

Errors occur because pixel values have to be in the range [0,255]. When 50 is subtracted from pixel values less than 50, the result is a negative value, which is an error. To prevent this from happening, two separate operations should be performed: 1) pixels with a value that is less than 50 should be set to zero 2) 50 should be subtracted from pixels with a value that is ¿= 50