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: Slower For Loops with Greyscale Image

```
import time
import skimage
from skimage import io
import numpy as np
import matplotlib.pyplot as plt
from skimage import img_as_float32
import sys

def process(file):
  A = io.imread(file)
  (m1,n1) = A.shape
  for i in range(m1):
    for j in range(n1):
      if A[i,j] <= 10 :
        A[i,j] = 0

start = time.time()
for file in sys.argv[1:]:
  process(file)
end = time.time()
print("total time: "+str(end - start))
```

Listing 2: Logical Indexing with Greyscale Image

```
import time
import skimage
from skimage import io
import numpy as np
import matplotlib.pyplot as plt
from skimage import img_as_float32
import sys

def process(file):
  A = io.imread(file)
  (m1,n1) = A.shape
  B = A < 10
  A[B] = 0

start = time.time()
for file in sys.argv[1:]:
  process(file)
end = time.time()
print("total time: "+str(end - start))
```

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

Listing 3: Command Line Performance Testing

```
gabe@dean:~/CSC-414_Project0$ python3 slow_gry.py dog.png
  grizzlypeakg.png heads.png mona_lisa.png yucca.png
total time: 5.245616674423218
gabe@dean:~/CSC-414_Project0$ python3 fast_gry.py dog.png
  grizzlypeakg.png heads.png mona_lisa.png yucca.png
total time: 0.0711512565612793
```

$$speedup = 5.245616674423218/0.0711512565612793$$

$$speedup = 73.724863452$$

(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.

Listing 4: Slower For Loops with Color Image

```
import time
import skimage
from skimage import io
import numpy as np
import matplotlib.pyplot as plt
from skimage import img_as_float32
import sys

def process(file):
  A = io.imread(file)
  (m1,n1,k1) = A.shape
  for i in range(m1):
    for j in range(n1):
      for k in range(k1):
        if A[i,j,k] <= 10 :
          A[i,j,k] = 0

start = time.time()
for file in sys.argv[1:]:
  process(file)
end = time.time()
print("total time: "+str(end - start))
```

Listing 5: Logical Indexing with Color Image

```
import time
import skimage
from skimage import io
import numpy as np
import matplotlib.pyplot as plt
from skimage import img_as_float32
import sys

def process(file):
  A = io.imread(file)
  (m1,n1,k1) = A.shape
  B = A < 10
  A[B] = 0

start = time.time()
for file in sys.argv[1:]:
  process(file)
end = time.time()
print("total time: "+str(end - start))
```

Listing 6: Command Line Performance Testing

```
gabe@dean:~/CSC-414_Project0$ python3 slow_rgb.py grizzlypeak.jpg
  total time: 11.221831321716309
gabe@dean:~/CSC-414_Project0$ python3 fast_rgb.py grizzlypeak.jpg
total time: 0.08472013473510742
  gabe@dean:~/CSC-414_Project0$
```

$$speedup = 11.221831321716309/0.08472013473510742$$

$$speedup = 132.457666135$$

We conclude that Color images get more speedup than Grayscale images since 132 is greater than 72.

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 greater than or equal to 50.