Total_variation_image_restoration

November 7, 2019

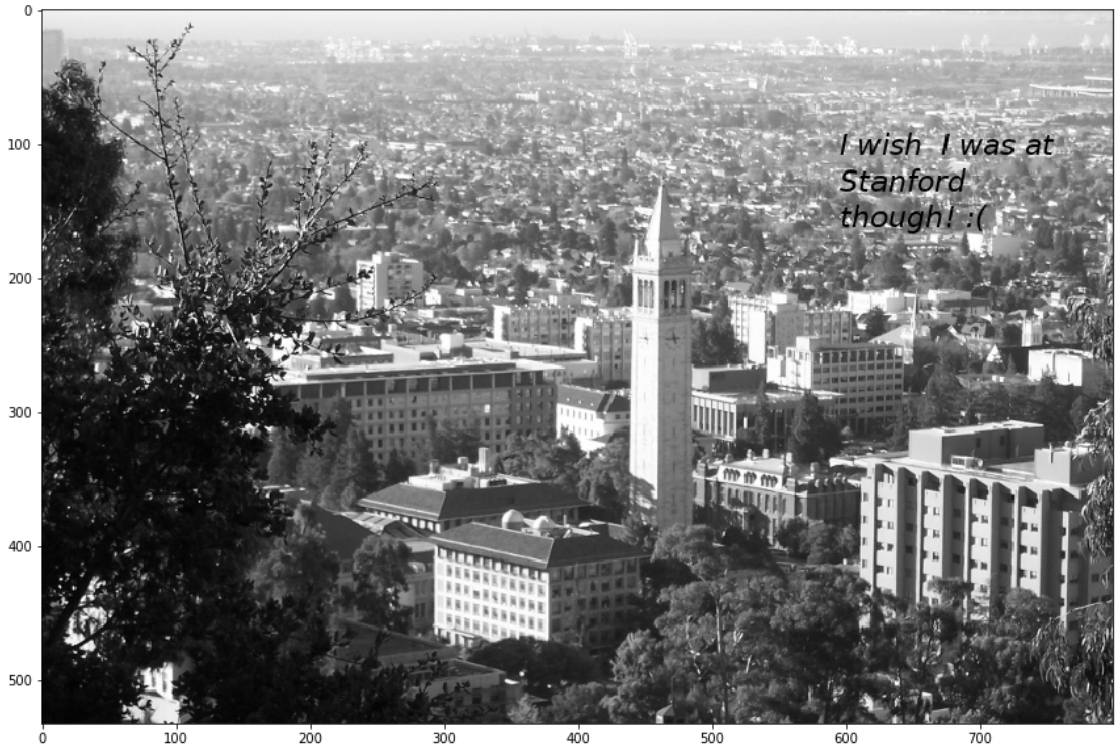
[3]:
```
import cv2
import cvxpy as cp
import numpy as np
import matplotlib.pyplot as plt
```

[4]:
```
# get image and constraints matrix
corrupted_image_filename = '../data/campanile_img_corrupted.jpg'
constraints_matrix_filename = '../data/constraint_matrix.txt'

u_corr = cv2.imread(corrupted_image_filename, 0)
F = u_corr

A = np.loadtxt(constraints_matrix_filename, delimiter=",")

# visualize image
fig = plt.figure(figsize=(30,10),facecolor='w')
ax = fig.add_subplot(111)
ax.imshow(u_corr, cmap='gray', vmin=0, vmax=255)
plt.show()
```



[9]:
```
rows, cols = F.shape
F_hat = cp.Variable(shape=(rows, cols))

# 1.5
# You complete this ---------------------------------------------------->
# obj = ?? (write objective function)
obj = cp.Minimize(cp.tv(F_hat))
# constraints = ?? (write constriants)
constraints = [cp.multiply(A, F) == cp.multiply(A, F_hat)]
prob = cp.Problem(obj, constraints)

# Use SCS to solve the problem.
# Could take about 10 mins to solve
prob.solve(verbose=True, solver=cp.SCS)
print("optimal objective value: {cv}".format(obj.value))
```

```
------------------------------------------------------------------------------
        SCS v2.1.1 - Splitting Conic Solver
        (c) Brendan O'Donoghue, Stanford University, 2012
------------------------------------------------------------------------------
Lin-sys: sparse-direct, nnz in A = 2549641
eps = 1.00e-04, alpha = 1.50, max_iters = 5000, normalize = 1, scale = 1.00
acceleration_lookback = 0, rho_x = 1.00e-03
```

```
Variables n = 851468, constraints m = 1701604
Cones:  primal zero / dual free vars: 426400
        soc vars: 1275204, soc blks: 425068
WARN: aa_init returned NULL, no acceleration applied.
Setup time: 5.81e+00s
------------------------------------------------------------------------------
 Iter | pri res | dua res | rel gap | pri obj | dua obj | kap/tau | time (s)
------------------------------------------------------------------------------
     0| 1.12e+22  1.35e+22  1.00e+00 -3.88e+29  2.22e+28  1.02e+29  3.48e-01
   100| 2.09e-03  1.19e-03  4.64e-04  8.30e+06  8.31e+06  3.71e-13  1.59e+01
   200| 7.77e-04  2.46e-04  1.34e-04  8.33e+06  8.33e+06  2.65e-09  3.13e+01
   300| 3.86e-04  1.12e-04  5.31e-05  8.34e+06  8.34e+06  2.67e-09  4.67e+01
   400| 1.92e-04  2.86e-05  2.18e-05  8.34e+06  8.34e+06  2.68e-09  6.21e+01
   500| 1.18e-04  5.40e-06  1.19e-05  8.34e+06  8.34e+06  2.68e-09  7.75e+01
   540| 9.92e-05  4.30e-06  9.56e-06  8.34e+06  8.34e+06  5.36e-09  8.37e+01
------------------------------------------------------------------------------
Status: Solved
Timing: Solve time: 8.37e+01s
        Lin-sys: nnz in L factor: 22215955, avg solve time: 8.54e-02s
        Cones: avg projection time: 4.59e-03s
        Acceleration: avg step time: 1.95e-07s
------------------------------------------------------------------------------
Error metrics:
dist(s, K) = 5.6843e-14, dist(y, K*) = 3.3307e-16, s'y/|s||y| = -1.0357e-18
primal res: |Ax + s - b|_2 / (1 + |b|_2) = 9.9215e-05
dual res:   |A'y + c|_2 / (1 + |c|_2) = 4.2952e-06
rel gap:    |c'x + b'y| / (1 + |c'x| + |b'y|) = 9.5562e-06
------------------------------------------------------------------------------
c'x = 8338290.4356, -b'y = 8338449.8023
==============================================================================


      ␣
 ↪------------------------------------------------------------------------------

       KeyError                                  Traceback (most recent call␣
 ↪last)

       <ipython-input-9-be3d44398493> in <module>
        13 # Could take about 10 mins to solve
        14 prob.solve(verbose=True, solver=cp.SCS)
   ---> 15 print("optimal objective value: {cv}".format(obj.value))


       KeyError: 'cv'
```

[10]:
```
# visualize result
fig = plt.figure(figsize=(30,10),facecolor='w')
ax = fig.add_subplot(111)
ax.imshow(F_hat.value, cmap='gray', vmin=0, vmax=255)
plt.show()
```

[ ]: