COMS 4735 Assignment 3 Write-up

Yang He

April 16th, 2018

Introduction

This assignment aims to use visual input (a map of Columbia campus) to generate textual descriptions of buildings and provide directions. There are three input files: new-campus.pgm (B/W image of the map), new-labeled.pgm (B/W image of the map with grayscale identifiers), and new-table.txt (a table of grayscale identifier values and building names).

The code listing is at: https://github.com/YHCU/ColumbiaMap.
All codes are original.

Step 0: Pre-Processing

The system will read in all three input files first, and store the two maps into 2D numpy arrays. Then the system reads in the value/name table and store them with two 1D arrays with matching indices, from 0 - 25 (26 buildings in total).

Step 1: Describing Shapes

Design Choices & Algorithms

The system will take the three input files and output 4 arrays: AREA (size of the building in pixel count), COM (center of mass in float coordinates), BBOX (max/min x and y of the building pixels), and DESC (shape descriptors). All these arrays can be inferred through matched indices. (i.e. index 0 stands for information of Pupin for all four arrays, as it is organized in new-table.txt).

AREA is calculated through counting the pixels with color value matching the value/name table for each of the building.

COM is calculated through diving the sum of x and y coordinates by the area of the building.

BBOX is calculated through finding the maximum and minimum of all the building pixel coordinates in both x and y direction.

Three types of shape descriptors are included: size, orientation and shape. No more descriptors are selected because most of the buildings are already identifiable through these descriptors with the exception of Mudd, and buildings with exactly the same shapes.

The size descriptor is given based on the building's size in comparison with the maximum size.

Size Descriptor	Size Ratio
LARGE	(0.67, 1]
MEDIUM	(0.33, 0.67]
SMALL	(0.1, 0.33]
TINY	(0, 0.1]

The orientation descriptor is given based on the ratio of bounding box's height and width.

Orientation Descriptor	Condition

VERTICAL	bbox_height > 1.25 * bbox_width
HORIZONTAL	bbox_width > 1.25 * bbox_height
SQUARELY ORIENTED	Others

There are two sets of shape descriptors as listed below:

1) SQUARE, RECTANGULAR, NON-RECTANGULAR

And if the building is NON-RECTANGULAR, the second set of descriptor is given.

2) SYMMETRICAL, NOT SYMMETRICAL ON BOTH ORIENTATIONS

The algorithm is designed as the following:

1) Compute the bounding box area.

2) If the bounding box area is smaller than 102% of the building area (2% is introduced to include floating errors), then it is either SQUARE or RECTANGULAR, go to step 3. Else, go to step 4.

3) If the bounding box width is within 10% of difference with the bounding box height, it is a SQUARE (10% accounts for not human noticable differences, buildings are highly unlikely designed to be exact square-shaped), or else it is RECTANGULAR.

4) The building is NON-RECTANGULAR. Compute the center coordinate of the bounding box.

5) If the center of mass is within 0.5% of difference with the bounding box center coordinate, the building is SYMMETRICAL, else it is NOT SYMMETRICAL ON BOTH ORIENTATIONS.

Output

Note that the sequence matches the sequence in new-table.txt.

AREA

```
array([1640., 1435., 5831., 1998., 5753., 3911., 3613.,  322., 1164.,
       1182., 1191., 3898., 1087.,  759., 1307., 1085.,  340.,  225.,
       1590., 1470., 4950., 2615., 5855., 2940., 5282., 1540.])
```

COM (x, y)

```
array([[ 76.15304878,  14.97621951],
       [143.        ,  20.        ],
       [223.24181101,  35.27954039],
       [ 16.        ,  40.5       ],
       [142.5256388 ,  99.46532244],
       [233.44694452, 120.51009972],
       [ 37.53086078, 119.54248547],
       [ 96.5       , 136.        ],
       [204.04896907, 175.98195876],
       [259.61505922, 176.        ],
       [ 17.        , 182.        ],
       [135.        , 221.5       ],
       [226.55381785, 222.02115915],
       [ 49.59947299, 221.8629776 ],
       [ 17.        , 259.        ],
       [258.33548387, 263.        ],
       [208.        , 253.5       ],
       [136.        , 276.        ],
       [ 41.5       , 301.09056604],
       [233.        , 300.9       ],
       [137.        , 322.5       ],
       [ 30.58699809, 363.94416826],
       [240.33885568, 416.94722459],
       [ 38.5       , 446.5       ],
       [132.        , 460.41461568],
       [ 38.5       , 479.5       ]])
```

BBOX (max_y, max_x, min_y, min_x)

```
array([[ 27., 115.,   3.,  39.],
       [ 37., 163.,   3., 123.],
       [ 86., 272.,   3., 166.],
       [ 77.,  29.,   4.,   3.],
       [147., 175.,  48., 110.],
       [147., 273.,  77., 181.],
       [147.,  80.,  81.,   3.],
       [147., 103., 125.,  90.],
       [201., 215., 151., 191.],
       [201., 272., 151., 247.],
       [206.,  31., 158.,   3.],
       [256., 169., 187., 101.],
       [234., 251., 210., 201.],
       [233.,  68., 211.,  31.],
       [285.,  31., 233.,   3.],
       [286., 272., 240., 245.],
       [261., 220., 246., 196.],
       [283., 143., 269., 129.],
       [311.,  80., 289.,   3.],
       [310., 272., 290., 194.],
       [331., 274., 314.,   0.],
       [414.,  81., 338.,   4.],
       [490., 270., 338., 191.],
       [467.,  73., 426.,   4.],
       [490., 179., 431.,  85.],
       [490.,  73., 469.,   4.]])
```

DESC

```
[['SMALL',
  'HORIZONTAL',
  'NON-RECTANGULAR',
  'NOT SYMMETRICAL ON BOTH ORIENTATION'],
 ['SMALL', 'SYMMETRICALLY ORIENTED', 'RECTANGULAR'],
 ['LARGE',
  'HORIZONTAL',
  'NON-RECTANGULAR',
  'NOT SYMMETRICAL ON BOTH ORIENTATION'],
 ['MEDIUM', 'VERTICAL', 'RECTANGULAR'],
 ['LARGE',
  'VERTICAL',
  'NON-RECTANGULAR',
  'NOT SYMMETRICAL ON BOTH ORIENTATION'],
 ['MEDIUM',
  'HORIZONTAL',
  'NON-RECTANGULAR',
  'NOT SYMMETRICAL ON BOTH ORIENTATION'],
 ['MEDIUM',
  'SYMMETRICALLY ORIENTED',
  'NON-RECTANGULAR',
  'NOT SYMMETRICAL ON BOTH ORIENTATION'],
 ['TINY', 'VERTICAL', 'RECTANGULAR'],
 ['SMALL',
  'VERTICAL',
  'NON-RECTANGULAR',
  'NOT SYMMETRICAL ON BOTH ORIENTATION'],
 ['SMALL', 'VERTICAL', 'NON-RECTANGULAR', 'SYMMETRICAL'],
 ['SMALL', 'VERTICAL', 'NON-RECTANGULAR', 'SYMMETRICAL'],
 ['MEDIUM', 'SYMMETRICALLY ORIENTED', 'NON-RECTANGULAR', 'SYMMETRICAL'],
 ['SMALL', 'HORIZONTAL', 'NON-RECTANGULAR', 'SYMMETRICAL'],
 ['SMALL', 'HORIZONTAL', 'NON-RECTANGULAR', 'SYMMETRICAL'],
 ['SMALL', 'VERTICAL', 'NON-RECTANGULAR', 'SYMMETRICAL'],
 ['SMALL', 'VERTICAL', 'NON-RECTANGULAR', 'SYMMETRICAL'],
 ['TINY', 'HORIZONTAL', 'NON-RECTANGULAR', 'SYMMETRICAL'],
 ['TINY', 'SMALLEST', 'SYMMETRICALLY ORIENTED', 'SQUARE'],
 ['SMALL', 'HORIZONTAL', 'NON-RECTANGULAR', 'SYMMETRICAL'],
 ['SMALL', 'HORIZONTAL', 'NON-RECTANGULAR', 'SYMMETRICAL'],
 ['LARGE', 'HORIZONTAL', 'RECTANGULAR'],
 ['MEDIUM',
  'SYMMETRICALLY ORIENTED',
  'NON-RECTANGULAR',
  'NOT SYMMETRICAL ON BOTH ORIENTATION'],
 ['LARGE',
  'LARGEST',
  'VERTICAL',
  'NON-RECTANGULAR',
  'NOT SYMMETRICAL ON BOTH ORIENTATION'],
 ['MEDIUM', 'HORIZONTAL', 'RECTANGULAR'],
 ['LARGE', 'HORIZONTAL', 'NON-RECTANGULAR', 'SYMMETRICAL'],
 ['SMALL', 'HORIZONTAL', 'RECTANGULAR']]
```

Step 2: Describing Absolute Space

Design Choices & Algorithms

To describe the absolute position of the buildings, two sets of descriptors are introduced.

1) NORTHWEST, NORTH, NORTHEAST, WEST, CENTER, EAST, SOUTHWEST, SOUTH, SOUTHEAST
2) UPPER, LOWER

The map is divided into a equally spaced 3x3 grid with the descriptor names. The first set of descriptor is based on the center of mass position of the building with regard to the grid, and assign the descriptor name to the building. The second set of descriptor is based on the center of mass position of the building within its grid - if it is above the grid center, it is UPPER, otherwise it is LOWER.

The reason for such design is to make the absolute space descriptors not too complicated while remain as distinct as possible for relative space computation purposes. It is not hard to notice that by observing the map, the center position of the buildings roughly fits into a 3x3 grid with the exception of the College Walk. The second set of descriptor will help to limit the maximum number of buildings with the same absolute space descriptors to 3 - which will make relative space computation much easier.

Output

```
[['NORTHWEST', 'UPPER'],
 ['NORTH', 'UPPER'],
 ['NORTHEAST', 'UPPER'],
 ['NORTHWEST', 'UPPER'],
 ['NORTH', 'LOWER'],
 ['NORTHEAST', 'LOWER'],
 ['NORTHWEST', 'LOWER'],
 ['NORTH', 'LOWER'],
 ['EAST', 'UPPER'],
 ['EAST', 'UPPER'],
 ['WEST', 'UPPER'],
 ['CENTER', 'UPPER'],
 ['EAST', 'UPPER'],
 ['WEST', 'UPPER'],
 ['WEST', 'LOWER'],
 ['EAST', 'LOWER'],
 ['EAST', 'LOWER'],
 ['CENTER', 'LOWER'],
 ['WEST', 'LOWER'],
 ['EAST', 'LOWER'],
 ['CENTER', 'LOWER'],
 ['SOUTHWEST', 'UPPER'],
 ['SOUTHEAST', 'LOWER'],
 ['SOUTHWEST', 'LOWER'],
 ['SOUTH', 'LOWER'],
 ['SOUTHWEST', 'LOWER']]
```

Step 3: Describing Relative Space Design Choices

For compass-directed spatial relationships (North, South, West, East) for (S, T), the relationship is marked true only when every pixel of S is in the same direction of every pixel of T, and it is determined through the max and min values of bounding box coordinates.

However, this might not be enough - for a pair of (S, T) that do not overlap each other at all with their bounding boxes, T could be, for example, both South and West of S. A test of their centers of mass is performed: if the line between the two centers are more vertical than horizontal, keep South and erase West, and vice versa. This would help to reduce every pair of (S, T) to only one compass-directed relationship, and it is more intuitive to human as well as easier to determine the path in later steps.

For Near relationships, I determined that the descriptor in Step 2 will first override any size considerations, simply because the grid is well-defined and very intuitive for near relationships, plus the fact that only considering size relationships would neglect some necessary ones - for example, alma mater is really near to the college walk; however, it is hard to determine that solely through the same numerical method applied to all buildings. After the override, a size relationship test is performed and is explained in the Algorithm part.

Algorithm

Given two buildings, the relative compass-directed spatial relationship (North, South, West, East) between the two is determined through the following way:

1) Check the COM of S and T: if the y difference is larger than x difference, then T is either North or South of S; otherwise, T is either East or West of S.
2) Check the max/min BBOX values of S and T: if T is North or South of S, when min_y_T > max_y_S, T is South of S, otherwise T is North of S; if T is East or West of S, when min_x_T > max_x_S, T is East of S, otherwise T is West of S.
3) Perform transitive reduction in the matrix. For any (i, j, k), if North(i, j) and North(j, k), set North(i, k) = false. Same goes for the other three matrices.
4) Perform reflective reduction in the matrices. For any (i, j), if North(i, j) = South(j, i) = True, set South(j, i) = False; if East(i, j) = West(j, i) = True, set West(j, i) = False.

The Near relationship is determined through the following way:

1) For every pair of buildings (i, j), if their descriptor in Step 2 is exactly the same, Near(i, j) = True.
2) For every other pair of buildings (i, j) that is not marked as true in 1), check the area within twice the scale of the bounding box of (i) around the center of mass of (i). If j appears in the area, Near(i, j) = True.
3) For all other unmarked (i, j) pairs, Near(i, j) = False.

Output

```
Pupin
--East-->SchapiroCEPSR

Pupin
--Near-->SchapiroCEPSR

Pupin
--Near-->NorthwestCorner

Pupin
--East-->Schermerhorn

Pupin
--East-->Fayerweather
```

```
SchapiroCEPSR
--Near-->Pupin

SchapiroCEPSR
--East-->Mudd&EngTerrace&Fairchild&CS

SchapiroCEPSR
--Near-->Mudd&EngTerrace&Fairchild&CS

SchapiroCEPSR
--Near-->Uris

Mudd&EngTerrace&Fairchild&CS
--Near-->SchapiroCEPSR

Mudd&EngTerrace&Fairchild&CS
--Near-->Uris

Mudd&EngTerrace&Fairchild&CS
--Near-->Schermerhorn

NorthwestCorner
--East-->Pupin

NorthwestCorner
--Near-->Pupin

NorthwestCorner
--East-->Uris

NorthwestCorner
--Near-->Chandler&Havemeyer

NorthwestCorner
--East-->Avery

NorthwestCorner
--East-->StPaulChapel

NorthwestCorner
--East-->Philosophy

Uris
--North-->Pupin

Uris
--Near-->Pupin

Uris
--North-->SchapiroCEPSR

Uris
--Near-->SchapiroCEPSR

Uris
--Near-->Mudd&EngTerrace&Fairchild&CS

Uris
--East-->Schermerhorn

Uris
--Near-->Schermerhorn
```

```
Uris
--Near-->Chandler&Havemeyer

Uris
--Near-->ComputerCenter

Uris
--Near-->Avery

Uris
--East-->Fayerweather

Uris
--Near-->LowLibrary

Schermerhorn
--North-->SchapiroCEPSR

Schermerhorn
--Near-->Mudd&EngTerrace&Fairchild&CS

Schermerhorn
--Near-->Uris

Schermerhorn
--Near-->Avery

Schermerhorn
--Near-->Fayerweather

Schermerhorn
--Near-->LowLibrary

Chandler&Havemeyer
--North-->Pupin

Chandler&Havemeyer
--East-->SchapiroCEPSR

Chandler&Havemeyer
--North-->NorthwestCorner

Chandler&Havemeyer
--Near-->NorthwestCorner

Chandler&Havemeyer
--Near-->Uris

Chandler&Havemeyer
--East-->ComputerCenter

Chandler&Havemeyer
--Near-->ComputerCenter

Chandler&Havemeyer
--Near-->Mathematics

Chandler&Havemeyer
--East-->Buell

Chandler&Havemeyer
--East-->Kent

ComputerCenter
```

```
--North-->Pupin

ComputerCenter
--North-->SchapiroCEPSR

ComputerCenter
--East-->Mudd&EngTerrace&Fairchild&CS

ComputerCenter
--North-->NorthwestCorner

ComputerCenter
--East-->Uris

ComputerCenter
--Near-->Uris

ComputerCenter
--East-->Avery

ComputerCenter
--East-->StPaulChapel

ComputerCenter
--East-->Philosophy

Avery
--North-->Mudd&EngTerrace&Fairchild&CS

Avery
--North-->Uris

Avery
--North-->Schermerhorn

Avery
--Near-->Schermerhorn

Avery
--East-->Fayerweather

Avery
--Near-->Fayerweather

Avery
--Near-->StPaulChapel

Fayerweather
--North-->Mudd&EngTerrace&Fairchild&CS

Fayerweather
--North-->Schermerhorn

Fayerweather
--Near-->Schermerhorn

Fayerweather
--Near-->Avery

Fayerweather
--Near-->StPaulChapel

Mathematics
--North-->SchapiroCEPSR
```

```
Mathematics
--North-->Chandler&Havemeyer

Mathematics
--Near-->Chandler&Havemeyer

Mathematics
--East-->ComputerCenter

Mathematics
--East-->LowLibrary

Mathematics
--Near-->EarlHall

Mathematics
--East-->AlmaMater

LowLibrary
--North-->Mudd&EngTerrace&Fairchild&CS

LowLibrary
--North-->Uris

LowLibrary
--North-->Schermerhorn

LowLibrary
--North-->Chandler&Havemeyer

LowLibrary
--North-->ComputerCenter

LowLibrary
--East-->Avery

LowLibrary
--Near-->Avery

LowLibrary
--East-->StPaulChapel

LowLibrary
--Near-->StPaulChapel

LowLibrary
--Near-->EarlHall

LowLibrary
--East-->Buell

LowLibrary
--Near-->Buell

LowLibrary
--Near-->AlmaMater

LowLibrary
--Near-->Dodge

LowLibrary
--East-->Kent
```

```
StPaulChapel
--North-->Avery

StPaulChapel
--Near-->Avery

StPaulChapel
--North-->Fayerweather

StPaulChapel
--Near-->Fayerweather

StPaulChapel
--Near-->Philosophy

EarlHall
--North-->Mudd&EngTerrace&Fairchild&CS

EarlHall
--North-->Uris

EarlHall
--East-->Schermerhorn

EarlHall
--North-->ComputerCenter

EarlHall
--North-->Mathematics

EarlHall
--Near-->Mathematics

EarlHall
--East-->LowLibrary

EarlHall
--Near-->Lewisohn

EarlHall
--East-->AlmaMater

Lewisohn
--North-->Mudd&EngTerrace&Fairchild&CS

Lewisohn
--North-->Uris

Lewisohn
--East-->Schermerhorn

Lewisohn
--North-->ComputerCenter

Lewisohn
--North-->Mathematics

Lewisohn
--East-->LowLibrary

Lewisohn
--Near-->EarlHall

Lewisohn
```

```
--East-->AlmaMater

Lewisohn
--Near-->Dodge

Lewisohn
--East-->Hamilton&Hartley&Wallach&JohnJay

Philosophy
--North-->StPaulChapel

Philosophy
--Near-->StPaulChapel

Philosophy
--Near-->Buell

Philosophy
--Near-->Kent

Buell
--North-->ComputerCenter

Buell
--North-->StPaulChapel

Buell
--East-->Philosophy

Buell
--Near-->Philosophy

Buell
--Near-->Kent

AlmaMater
--North-->Avery

AlmaMater
--East-->Fayerweather

AlmaMater
--North-->LowLibrary

AlmaMater
--East-->StPaulChapel

AlmaMater
--East-->Buell

AlmaMater
--East-->Kent

AlmaMater
--Near-->CollegeWalk

Dodge
--East-->Schermerhorn

Dodge
--East-->LowLibrary

Dodge
--North-->EarlHall
```

Dodge
--North-->Lewisohn

Dodge
--Near-->Lewisohn

Dodge
--East-->AlmaMater

Dodge
--Near-->CollegeWalk

Dodge
--East-->Hamilton&Hartley&Wallach&JohnJay

Kent
--North-->Philosophy

Kent
--Near-->Philosophy

Kent
--North-->Buell

Kent
--Near-->Buell

Kent
--Near-->CollegeWalk

CollegeWalk
--North-->StPaulChapel

CollegeWalk
--North-->EarlHall

CollegeWalk
--North-->AlmaMater

CollegeWalk
--Near-->AlmaMater

CollegeWalk
--Near-->Dodge

CollegeWalk
--Near-->Kent

CollegeWalk
--Near-->Journalism&Fourfold

CollegeWalk
--Near-->Hamilton&Hartley&Wallach&JohnJay

Journalism&Fourfold
--North-->Avery

Journalism&Fourfold
--North-->LowLibrary

Journalism&Fourfold
--East-->AlmaMater

```
Journalism&Fourfold
--North-->Dodge

Journalism&Fourfold
--Near-->Dodge

Journalism&Fourfold
--Near-->CollegeWalk

Journalism&Fourfold
--Near-->Lerner

Journalism&Fourfold
--East-->ButlerLibrary

Journalism&Fourfold
--Near-->ButlerLibrary

Hamilton&Hartley&Wallach&JohnJay
--North-->EarlHall

Hamilton&Hartley&Wallach&JohnJay
--Near-->Philosophy

Hamilton&Hartley&Wallach&JohnJay
--North-->AlmaMater

Hamilton&Hartley&Wallach&JohnJay
--North-->Kent

Hamilton&Hartley&Wallach&JohnJay
--Near-->Kent

Hamilton&Hartley&Wallach&JohnJay
--Near-->CollegeWalk

Hamilton&Hartley&Wallach&JohnJay
--Near-->ButlerLibrary

Lerner
--East-->Philosophy

Lerner
--North-->Buell

Lerner
--East-->Kent

Lerner
--North-->CollegeWalk

Lerner
--North-->Journalism&Fourfold

Lerner
--Near-->Journalism&Fourfold

Lerner
--East-->ButlerLibrary

Lerner
--Near-->ButlerLibrary

Lerner
```

```
--Near-->Carman

ButlerLibrary
--North-->Dodge

ButlerLibrary
--North-->Kent

ButlerLibrary
--North-->CollegeWalk

ButlerLibrary
--East-->Hamilton&Hartley&Wallach&JohnJay

ButlerLibrary
--Near-->Hamilton&Hartley&Wallach&JohnJay

ButlerLibrary
--Near-->Lerner

ButlerLibrary
--Near-->Carman

Carman
--East-->Philosophy

Carman
--East-->Kent

Carman
--North-->Lerner

Carman
--Near-->Lerner

Carman
--East-->ButlerLibrary

Carman
--Near-->ButlerLibrary
```