Natural Language and Image Description

Federick Gonzalez fag2113

April 2020

Abstract

In this assignment, an attempt was made to describe a visual image in human language terms. A 2-D, black and white image of Columbia's campus was used, along with a "Source" and a "Target". Descriptions of both of these locations were printed in non-numerical terms, along with instructions on how to get from the source to the target, without the utilization of building names.

1 "What": Describing shape

In order to determine the shape of an object, the following 4 categories were created:

- Small, Medium, or Large.
- Has Holes, or has No Holes.
- Is Symmetric or is Not Symmetric.
- Square, Rectangular, Complicated Square, Complicated Rectangle, LShaped, CShaped, or IShaped.

In order to determine the size, small, medium, or large. the areas of all 26 shapes were compared. It was deemed that something roughly smaller than 1000 pixels would be considered small, anything that is Smaller than 3000 pixels is Medium, and anything larger than that is considered Large. These values were obtained by asking users whether or not a highlighted building was small, medium, or large, and then fine-tuning the pixel values to match this.

In order to determine a shape's symmetry, a mask was created over the original campus map containing only the current building. That map was reflected over the X axis, and then the map and it's reflection had contours discovered. These contours were compared, and if their difference was less than 50%, then they were symmetric. The similarities between contours were calculated using OpenCV's matchShapes method.

In order to determine whether or not an object has holes, OpenCV's contour hierarchy was formed. The contour hierarchy allows for the program to

Figure 1: An example of an L-Shaped building

detect whether a shape is wholly within another shape. The first hierarchy level therefore becomes the outer bounds of a shape, and the second hierarchy level becomes the inner bounds of a shape. By checking if the next discovered contour was in the second hierarchy level, it was possible to determine whether or not a certain building had holes in it.

In order to determine shape, the number of edges for a building's countours were calculated. If there were 4 edges, then it was either a square or a rectangle, based on the proportion of width to height. If there were more edges, the minimum bounding rectangle surrounding a building was converted into an array, with 0's being empty space.

To check if a building was LShaped, each array was checked for columns in which there was only one run of non-zero values. A run means that once the first non-zero value is discovered, all non-zero values discovered until the next zero is considered a run. If the shape is an L, then a majority of the columns should contain runs, as an L shape is completely solid, with no gaps between a filled in area and another filled in area above it. However, there were some buildings that were not completely solid L-shapes, therefore through some testing it was discovered that if more than $\frac{17}{24}$ columns contained only one run, then that object could be considered an L. An example of a building that can be considered L-shaped can be seen in figure 1.

To check if a building was C-shaped, the columns were analyzed to find where there were 2 runs. If the amount of columns with only 2 runs is greater than $\frac{2}{3}$ of the entire array, then the shape was determined to be a C. However, because a C could be both vertical and horizontal, checking for runs was performed both along the rows and along the columns, using similar methods. However, the horizontal C-shape searched for runs greater than $\frac{1}{3}$, as the horizontal C-shapes extensions were significantly smaller than those of the vertical c-shapes. Examples of vertical C shapes and horizontal C shapes can be found in figures 2 and 3, respectively.

Finally, to check if a building was I-shaped, almost the same process was used as checking for C-shapes, except now the number of columns with only 2

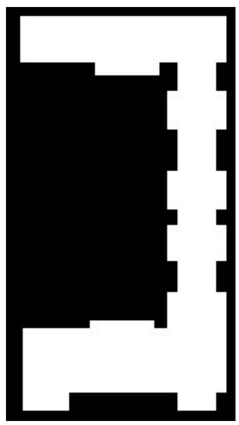

Figure 2: One example of a C shape

Figure 3: Another C shape

runs per column was limited to $\frac{1}{3}$. This is mainly because in those buildings that were visually I-Shaped, The extensions along the top and bottom of the I shape were quite small, and therefore were assumed to be half as large as the extensions along the top and bottom of the vertical C-shape. An example of an I shaped building can be found in figure 4.

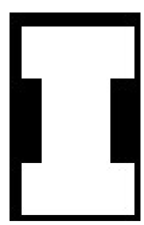

Figure 4: An example of an I Shaped building

Included below is an image of the outputs of "whats" generated by the 26 buildings. The details included follow the order:
[Building Name, Center of Mass, Area, Top left corner of MBR, Bottom Right Corner of MBR, size, shape, symmetric, hasHoles]

```
['Carman', (38, 479), 1542.6365625000003, (3.8500000000000005,
    468.87500000000006), (74.25, 491.15000000000003), 'medium',
    'rectangular', 'isSymmetric', 'hasNoHoles']
['ButlerLibrary', (132, 460), 5312.429375000001, (84.7,
    430.65000000000003), (180.125, 491.15000000000003), 'large',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles']
['Lerner', (38, 446), 2969.2643750000007, (3.8500000000000005,
    425.70000000000005), (74.25, 468.32500000000005), 'medium',
    'rectangular', 'isSymmetric', 'hasNoHoles']
```

```
['Hamilton&Hartley&Wallach&JohnJay', (240, 417),
    5915.992500000001, (190.85000000000002, 337.70000000000005),
    (271.15000000000003, 491.1500000000001), 'large', 'CShaped',
    'isNotSymmetric', 'hasNoHoles']
['Journalism&Furnald', (30, 364), 2649.6731250000003,
    (3.8500000000000005, 337.70000000000005), (82.225,
    415.25000000000006), 'medium', 'LShaped', 'isNotSymmetric',
    'hasNoHoles']
['CollegeWalk', (137, 322), 4986.258750000001, (0.0,
    313.77500000000003), (275.0, 332.20000000000005), 'large',
    'rectangular', 'isSymmetric', 'hasNoHoles']
['Kent', (233, 301), 1493.1400000000003, (193.87500000000003,
    289.85), (273.35, 311.3), 'medium', 'CShaped', 'isSymmetric',
    'hasNoHoles']
['Dodge', (41, 301), 1599.3553125000003, (2.75, 288.75), (81.125,
    312.125), 'medium', 'CShaped', 'isSymmetric', 'hasNoHoles']
['AlmaMater', (136, 276), 232.32000000000005,
    (128.70000000000002, 268.675), (144.10000000000002, 284.35),
    'small', 'square', 'isSymmetric', 'hasNoHoles']
['Buell', (208, 253), 345.75750000000005, (195.8,
    245.85000000000002), (221.10000000000002, 262.35), 'small',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles']
['Philosophy', (258, 263), 1101.5915625000002,
    (244.75000000000003, 239.8), (273.35, 287.1), 'medium',
    'IShaped', 'isSymmetric', 'hasNoHoles']
['Lewisohn', (17, 259), 1328.655625, (2.75, 232.65), (32.175,
    286.27500000000003), 'medium', 'IShaped', 'isSymmetric',
    'hasNoHoles']
['EarlHall', (49, 222), 774.0975000000001, (30.800000000000004,
    210.65), (69.30000000000001, 234.3), 'small',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles']
['StPaulChapel', (226, 222), 1096.9784375000002,
    (200.75000000000003, 209.82500000000002),
    (252.17500000000004, 235.12500000000003), 'medium',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles']
['LowLibrary', (135, 221), 3922.7443750000007, (100.65,
    186.72500000000002), (170.22500000000002, 257.125), 'large',
    'complicatedSquare', 'isSymmetric', 'hasNoHoles']
['Mathematics', (17, 182), 1208.4875000000002, (2.75,
    157.85000000000002), (32.175, 207.35000000000002), 'medium',
    'IShaped', 'isSymmetric', 'hasNoHoles']
['Fayerweather', (259, 176), 1200.8493750000002, (246.675,
    150.70000000000002), (273.35, 202.12500000000003), 'medium',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles']
['Avery', (204, 176), 1167.8012500000002, (190.85000000000002,
    150.70000000000002), (216.15000000000003,
```

```
    202.12500000000003), 'medium', 'complicatedRectangle',
    'isSymmetric', 'hasNoHoles']
['OldComputerCenter', (96, 136), 329.83843750000005, (89.65,
    124.85000000000001), (104.22500000000001,
    148.22500000000002), 'small', 'rectangular', 'isSymmetric',
    'hasNoHoles']
['Chandler&Havemeyer', (37, 119), 3638.4700000000007, (2.75,
    80.85000000000001), (81.125, 148.22500000000002), 'large',
    'LShaped', 'isSymmetric', 'hasNoHoles']
['Schermerhorn', (233, 120), 3951.9356250000005, (180.675,
    76.72500000000001), (274.175, 148.22500000000002), 'large',
    'CShaped', 'isSymmetric', 'hasNoHoles']
['Uris', (142, 99), 5780.888437500001, (109.72500000000001,
    47.85), (176.27500000000003, 148.22500000000002), 'large',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles']
['NorthwestCorner', (16, 40), 2013.8181250000002, (2.75,
    3.8500000000000005), (30.250000000000004, 78.1), 'medium',
    'rectangular', 'isSymmetric', 'hasNoHoles']
['Mudd&EngTerrace&Fairchild&CS', (224, 35), 6144.001875000001,
    (165.82500000000002, 2.75), (273.35, 87.17500000000001),
    'large', 'IShaped', 'isSymmetric', 'hasHoles']
['SchapiroCEPSR', (143, 20), 1451.4706250000002, (122.65, 2.75),
    (164.175, 38.225), 'medium', 'rectangular', 'isSymmetric',
    'hasNoHoles']
['Pupin', (76, 15), 1662.5778125000002, (38.775000000000006,
    2.75), (116.32500000000002, 28.325000000000003), 'medium',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles']
```

2 "Where": Describing absolute space

To determine the position of an object, the following categories were used:

- hasNoSingleOrientation, orientedNorth2South, orientedEast2West
- notNorthmostNorSouthmost, onNorthBorder, onSouthBorder
- notEastmostNorWestmost , onEastBorder , onWestBorder
- East Half , West half
- North Half, South Half

In order to determine a shape's orientation, OpenCV was used to fit an ellipse to the building's contours. Due to this ellipse, the angle of the building's central axis could be determined, since it is perpendicular to the main axis of the bounding ellipse. If the ellipse angle was near 0 or near 180, this meant that the building was orientedNorth2South. If the angle was near 90 degrees, the building was checked to see whether or not it was a square. If it was, then it had no single orientation, otherwise it was orientedEast2West.

In order to determine what border each building was on, the minimum bounding rectangle was computed. If the top left corner was either within 7 pixels of the top left edges of the image, where Y = 0 or o = 0, then it was determined to be on the north border or the west border, respectively. Using the width and height of the minimum bounding rectangle, the location of the bottom left corner was calculated. If it was within 15 pixels of the bottom or right edges, where Y = 495 of X = 275, the shape was determined to be on the south or east border, respectively. The pixel distance from the edges were a little bit larger on the southernmost and easternmost edges because through analysis of the pgm files, it was discovered that there was extra white space along these two edges compared to the northern or western edges.

In order to determine which half the building lay in, the center of mass was calculated and if the X coordinate was less than half of the width of the image, it fell in the west half, otherwise it was in the east half. Similarly, if the Y coordinate was less than half of the height of the image, it fell into the north half, otherwise it fell into the southern half.

Included below is an image of the outputs of "whats and wheres" generated by the 26 buildings. The details included follow the order:
[Building Name, Center of Mass, Area, Top left corner of MBR, Bottom Right Corner of MBR, size, shape, symmetric, hasHoles, orientation, borders, location]

```
['Carman', (38, 479), 1542.6365625000003, (3.8500000000000005,
    468.87500000000006), (74.25, 491.15000000000003), 'medium',
    'rectangular', 'isSymmetric', 'hasNoHoles',
    'orientedEast2West', ['onSouthBorder', 'onWestBorder'],
    'WestHalf', 'SouthHalf']
['ButlerLibrary', (132, 460), 5312.429375000001, (84.7,
    430.65000000000003), (180.125, 491.15000000000003), 'large',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles',
    'orientedEast2West', ['onSouthBorder',
    'NotEasternmostNorWesternmost'], 'WestHalf', 'SouthHalf']
['Lerner', (38, 446), 2969.2643750000007, (3.8500000000000005,

    425.70000000000005), (74.25, 468.32500000000005), 'medium',
    'rectangular', 'isSymmetric', 'hasNoHoles', 'orientedEast2West',
    ['NotNorternmostNorSouthernmost', 'onWestBorder'], 'WestHalf',
    'SouthHalf']
['Hamilton&Hartley&Wallach&JohnJay', (240, 417),
    5915.992500000001, (190.85000000000002, 337.70000000000005),
    (271.15000000000003, 491.1500000000001), 'large', 'CShaped',
    'isNotSymmetric', 'hasNoHoles', 'orientedNorth2South',
    ['onSouthBorder', 'onEastBorder'], 'EastHalf', 'SouthHalf']
```

```
['Journalism&Furnald', (30, 364), 2649.6731250000003,
    (3.8500000000000005, 337.70000000000005), (82.225,
    415.25000000000006), 'medium', 'LShaped', 'isNotSymmetric',
    'hasNoHoles', 'noSingleOrientation',
    ['NotNorternmostNorSouthernmost', 'onWestBorder'],
    'WestHalf', 'SouthHalf']
['CollegeWalk', (137, 322), 4986.258750000001, (0.0,
    313.77500000000003), (275.0, 332.20000000000005), 'large',
    'rectangular', 'isSymmetric', 'hasNoHoles',
    'orientedEast2West', ['NotNorternmostNorSouthernmost',
    'onEastBorder', 'onWestBorder'], 'WestHalf', 'SouthHalf']
['Kent', (233, 301), 1493.1400000000003, (193.87500000000003,
    289.85), (273.35, 311.3), 'medium', 'CShaped', 'isSymmetric',
    'hasNoHoles', 'orientedEast2West',
    ['NotNorternmostNorSouthernmost', 'onEastBorder'],
    'EastHalf', 'SouthHalf']
['Dodge', (41, 301), 1599.3553125000003, (2.75, 288.75), (81.125,
    312.125), 'medium', 'CShaped', 'isSymmetric', 'hasNoHoles',
    'orientedEast2West', ['NotNorternmostNorSouthernmost',
    'onWestBorder'], 'WestHalf', 'SouthHalf']
['AlmaMater', (136, 276), 232.32000000000005,
    (128.70000000000002, 268.675), (144.10000000000002, 284.35),
    'small', 'square', 'isSymmetric', 'hasNoHoles',
    'NoSingleOrientation', ['NotNorternmostNorSouthernmost',
    'NotEasternmostNorWesternmost'], 'WestHalf', 'SouthHalf']
['Buell', (208, 253), 345.75750000000005, (195.8,
    245.85000000000002), (221.10000000000002, 262.35), 'small',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles',
    'orientedEast2West', ['NotNorternmostNorSouthernmost',
    'NotEasternmostNorWesternmost'], 'EastHalf', 'SouthHalf']
['Philosophy', (258, 263), 1101.5915625000002,
    (244.75000000000003, 239.8), (273.35, 287.1), 'medium',
    'IShaped', 'isSymmetric', 'hasNoHoles',
    'orientedNorth2South', ['NotNorternmostNorSouthernmost',
    'onEastBorder'], 'EastHalf', 'SouthHalf']
['Lewisohn', (17, 259), 1328.655625, (2.75, 232.65), (32.175,
    286.27500000000003), 'medium', 'IShaped', 'isSymmetric',
    'hasNoHoles', 'orientedNorth2South',
    ['NotNorternmostNorSouthernmost', 'onWestBorder'],
    'WestHalf', 'SouthHalf']
['EarlHall', (49, 222), 774.0975000000001, (30.800000000000004,
    210.65), (69.30000000000001, 234.3), 'small',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles',
    'orientedEast2West', ['NotNorternmostNorSouthernmost',
    'NotEasternmostNorWesternmost'], 'WestHalf', 'NorthHalf']
['StPaulChapel', (226, 222), 1096.9784375000002,
```

```
    (200.75000000000003, 209.82500000000002),
    (252.17500000000004, 235.12500000000003), 'medium',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles',
    'orientedEast2West', ['NotNorternmostNorSouthernmost',
    'NotEasternmostNorWesternmost'], 'EastHalf', 'NorthHalf']
['LowLibrary', (135, 221), 3922.7443750000007, (100.65,
    186.72500000000002), (170.22500000000002, 257.125), 'large',
    'complicatedSquare', 'isSymmetric', 'hasNoHoles',
    'noSingleOrientation', ['NotNorternmostNorSouthernmost',
    'NotEasternmostNorWesternmost'], 'WestHalf', 'NorthHalf']
['Mathematics', (17, 182), 1208.4875000000002, (2.75,
    157.85000000000002), (32.175, 207.35000000000002), 'medium',
    'IShaped', 'isSymmetric', 'hasNoHoles',
    'orientedNorth2South', ['NotNorternmostNorSouthernmost',
    'onWestBorder'], 'WestHalf', 'NorthHalf']
['Fayerweather', (259, 176), 1200.8493750000002, (246.675,
    150.70000000000002), (273.35, 202.12500000000003), 'medium',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles',
    'orientedNorth2South', ['NotNorternmostNorSouthernmost',
    'onEastBorder'], 'EastHalf', 'NorthHalf']
['Avery', (204, 176), 1167.8012500000002, (190.85000000000002,
    150.70000000000002), (216.15000000000003,
    202.12500000000003), 'medium', 'complicatedRectangle',
    'isSymmetric', 'hasNoHoles', 'orientedNorth2South',
    ['NotNorternmostNorSouthernmost',
    'NotEasternmostNorWesternmost'], 'EastHalf', 'NorthHalf']
['OldComputerCenter', (96, 136), 329.83843750000005, (89.65,
    124.85000000000001), (104.22500000000001,
    148.22500000000002), 'small', 'rectangular', 'isSymmetric',
    'hasNoHoles', 'orientedNorth2South',
    ['NotNorternmostNorSouthernmost',
    'NotEasternmostNorWesternmost'], 'WestHalf', 'NorthHalf']
['Chandler&Havemeyer', (37, 119), 3638.4700000000007, (2.75,
    80.85000000000001), (81.125, 148.22500000000002), 'large',
    'LShaped', 'isSymmetric', 'hasNoHoles',
    'noSingleOrientation', ['NotNorternmostNorSouthernmost',
    'onWestBorder'], 'WestHalf', 'NorthHalf']
['Schermerhorn', (233, 120), 3951.9356250000005, (180.675,
    76.72500000000001), (274.175, 148.22500000000002), 'large',
    'CShaped', 'isSymmetric', 'hasNoHoles',
    'noSingleOrientation', ['NotNorternmostNorSouthernmost',
    'onEastBorder'], 'EastHalf', 'NorthHalf']
['Uris', (142, 99), 5780.888437500001, (109.72500000000001,
    47.85), (176.27500000000003, 148.22500000000002), 'large',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles',
    'orientedNorth2South', ['NotNorternmostNorSouthernmost',
```

```
    'NotEasternmostNorWesternmost'], 'EastHalf', 'NorthHalf']
['NorthwestCorner', (16, 40), 2013.8181250000002, (2.75,
    3.8500000000000005), (30.250000000000004, 78.1), 'medium',
    'rectangular', 'isSymmetric', 'hasNoHoles',
    'orientedNorth2South', ['onNorthBorder', 'onWestBorder'],
    'WestHalf', 'NorthHalf']
['Mudd&EngTerrace&Fairchild&CS', (224, 35), 6144.001875000001,
    (165.82500000000002, 2.75), (273.35, 87.17500000000001),
    'large', 'IShaped', 'isSymmetric', 'hasHoles',
    'noSingleOrientation', ['onNorthBorder', 'onEastBorder'],
    'EastHalf', 'NorthHalf']
['SchapiroCEPSR', (143, 20), 1451.4706250000002, (122.65, 2.75),
    (164.175, 38.225), 'medium', 'rectangular', 'isSymmetric',
    'hasNoHoles', 'orientedNorth2South', ['onNorthBorder',
    'NotEasternmostNorWesternmost'], 'EastHalf', 'NorthHalf']
['Pupin', (76, 15), 1662.5778125000002, (38.775000000000006,
    2.75), (116.32500000000002, 28.325000000000003), 'medium',
    'complicatedRectangle', 'isSymmetric', 'hasNoHoles',
    'orientedEast2West', ['onNorthBorder',
    'NotEasternmostNorWesternmost'], 'WestHalf', 'NorthHalf']
```

3 "How": Describing relative space

In order to identify spatial relationships, it was necessary to determine whether each building was North, South, East, West, or near each other building. North was determined by comparing the top border of one shape to the bottom border of another shape. If the top border of a source was below the bottom border of a target, the direction from source to target is north, and vice versa. The same occurred for east and west, except using the left and right borders. In order to keep track of these relationships, 5 26x26 binary matrices were produced containing all the possible relationships. This however was a large amount of information and it needed to be trimmed down. To trim down the amount of data in the directional matrices matrix, For each building, the directional relations were compared. For example, using only the North matrix, if a building C is north of A, and a building C is north of A and B, then it can be removed from the list of buildings to the north of A, since C is north of B which is north of A.

The near description was a little bit more complicated. Since each building was to be viewed as a blob, each building was converted into 9 points: Each corner, the midpoint of each edge, and then the center. If the distance between any point on one object to any point on another was less than a maximum distance, then the two objects were determined to be near to each other. The maximum distance was discovered by asking users to select buildings they thought were close to each other, and then editing the maximum distance value until the buildings deemed close to each other were the relations returned by the Near method.

The spatial relationships after trimming can be found below:

```
Lerner north Carman
Journalism&Furnald north ButlerLibrary
Journalism&Furnald north Lerner
Journalism&Furnald north Hamilton&Hartley&Wallach&JohnJay
CollegeWalk north Journalism&Furnald
Kent north CollegeWalk
AlmaMater north Kent
AlmaMater north Dodge
Buell north AlmaMater
EarlHall north Buell
Buell north Philosophy
EarlHall north Lewisohn
Mathematics north EarlHall
Mathematics north StPaulChapel
Mathematics north LowLibrary
OldComputerCenter north Mathematics
OldComputerCenter north Fayerweather
OldComputerCenter north Avery
NorthwestCorner north OldComputerCenter
NorthwestCorner north Chandler&Havemeyer
NorthwestCorner north Schermerhorn
NorthwestCorner north Uris
SchapiroCEPSR north NorthwestCorner
Pupin north Mudd&EngTerrace&Fairchild&CS
Carman south Lerner
Lerner south Journalism&Furnald
Journalism&Furnald south CollegeWalk
CollegeWalk south Kent
CollegeWalk south Dodge
Dodge south AlmaMater
AlmaMater south Buell
Dodge south Philosophy
Dodge south Lewisohn
Philosophy south EarlHall
Philosophy south StPaulChapel
AlmaMater south LowLibrary
StPaulChapel south Mathematics
StPaulChapel south Fayerweather
StPaulChapel south Avery
Avery south OldComputerCenter
Avery south Chandler&Havemeyer
```

```
Avery south Schermerhorn
Avery south Uris
Chandler&Havemeyer south NorthwestCorner
OldComputerCenter south Mudd&EngTerrace&Fairchild&CS
Uris south SchapiroCEPSR
Uris south Pupin
ButlerLibrary and AlmaMater and LowLibrary and OldComputerCenter east Carman
Hamilton&Hartley&Wallach&JohnJay and Kent and Buell and StPaulChapel
    and Avery and Schermerhorn east ButlerLibrary
ButlerLibrary and AlmaMater and LowLibrary and OldComputerCenter
    east Lerner
ButlerLibrary and AlmaMater and LowLibrary and OldComputerCenter
    east Journalism&Furnald
ButlerLibrary and AlmaMater and LowLibrary and OldComputerCenter east Dodge
Hamilton&Hartley&Wallach&JohnJay and Kent and Buell and StPaulChapel and
    Avery and Schermerhorn and Mudd&EngTerrace&Fairchild&CS east AlmaMater
Philosophy and Fayerweather east Buell
ButlerLibrary and AlmaMater and Philosophy and LowLibrary and OldComputerCenter
    and Pupin east Lewisohn
ButlerLibrary and AlmaMater and Philosophy and LowLibrary and OldComputerCenter
    east EarlHall
Hamilton&Hartley&Wallach&JohnJay and Kent and Buell and StPaulChapel and
    Avery and Schermerhorn east LowLibrary
ButlerLibrary and AlmaMater and Philosophy and LowLibrary and Fayerweather
    and OldComputerCenter and Pupin east Mathematics
Philosophy and Fayerweather east Avery
Hamilton&Hartley&Wallach&JohnJay and Kent and AlmaMater and
    Philosophy and Fayerweather and Uris and SchapiroCEPSR east
    OldComputerCenter
ButlerLibrary and AlmaMater and Philosophy and LowLibrary and
    Fayerweather and OldComputerCenter east Chandler&Havemeyer
Hamilton&Hartley&Wallach&JohnJay and Kent and Buell and
    StPaulChapel and Avery and Schermerhorn east Uris
ButlerLibrary and AlmaMater and Philosophy and EarlHall and
    Fayerweather and Uris and SchapiroCEPSR and Pupin east NorthwestCorner
Hamilton&Hartley&Wallach&JohnJay and Kent and Buell and
    StPaulChapel and Avery and Schermerhorn and
    Mudd&EngTerrace&Fairchild&CS east SchapiroCEPSR
Hamilton&Hartley&Wallach&JohnJay and Kent and AlmaMater and
    Philosophy and Fayerweather and SchapiroCEPSR east Pupin
Carman and Lerner and Journalism&Furnald and Dodge and Lewisohn
    and EarlHall and Mathematics and Chandler&Havemeyer and
    NorthwestCorner west ButlerLibrary
Carman and ButlerLibrary and AlmaMater and LowLibrary and Uris
    and SchapiroCEPSR and Pupin west
    Hamilton&Hartley&Wallach&JohnJay
```

```
Carman and ButlerLibrary and AlmaMater and LowLibrary and Uris
    and SchapiroCEPSR and Pupin west Kent
Carman and OldComputerCenter and Chandler&Havemeyer and Pupin
    west AlmaMater
Carman and ButlerLibrary and AlmaMater and LowLibrary and Uris
    and SchapiroCEPSR and Pupin west Buell
Carman and Buell and Avery and Uris and SchapiroCEPSR and Pupin
    west Philosophy
NorthwestCorner west EarlHall
Carman and ButlerLibrary and AlmaMater and LowLibrary and Uris
    and SchapiroCEPSR and Pupin west StPaulChapel
Carman and Lerner and Journalism&Furnald and Dodge and Lewisohn
    and EarlHall and Mathematics and Chandler&Havemeyer and
    NorthwestCorner west LowLibrary
Carman and Buell and Avery and Uris and SchapiroCEPSR and Pupin
    west Fayerweather
Carman and ButlerLibrary and AlmaMater and LowLibrary and Uris
    and SchapiroCEPSR and Pupin west Avery
Carman and Lerner and Journalism&Furnald and Dodge and Lewisohn
    and EarlHall and Mathematics and Chandler&Havemeyer and
    NorthwestCorner west OldComputerCenter
Carman and ButlerLibrary and AlmaMater and LowLibrary and Uris
    and SchapiroCEPSR and Pupin west Schermerhorn
Carman and OldComputerCenter and Chandler&Havemeyer and
    NorthwestCorner west Uris
Carman and AlmaMater and SchapiroCEPSR and Pupin west
    Mudd&EngTerrace&Fairchild&CS
Carman and OldComputerCenter and Chandler&Havemeyer and Pupin
    west SchapiroCEPSR
Lewisohn and Mathematics and NorthwestCorner west Pupin
ButlerLibrary and Lerner near Carman
Carman and Lerner and Hamilton&Hartley&Wallach&JohnJay and
    Journalism&Furnald near ButlerLibrary
Carman and ButlerLibrary and Journalism&Furnald near Lerner
ButlerLibrary and CollegeWalk and Kent near
    Hamilton&Hartley&Wallach&JohnJay
ButlerLibrary and Lerner and CollegeWalk and Dodge near
    Journalism&Furnald
Hamilton&Hartley&Wallach&JohnJay and Journalism&Furnald and Kent
    and Dodge and AlmaMater and Philosophy and Lewisohn near
    CollegeWalk
Hamilton&Hartley&Wallach&JohnJay and CollegeWalk and Buell and
    Philosophy and LowLibrary near Kent
Journalism&Furnald and CollegeWalk and Lewisohn and LowLibrary
    near Dodge
CollegeWalk and LowLibrary near AlmaMater
```

```
Kent and Philosophy and StPaulChapel and LowLibrary and Avery
    near Buell
CollegeWalk and Kent and Buell and StPaulChapel and Fayerweather
    near Philosophy
CollegeWalk and Dodge and EarlHall and Mathematics near Lewisohn
Lewisohn and LowLibrary and Mathematics near EarlHall
Buell and Philosophy and LowLibrary and Fayerweather and Avery
    near StPaulChapel
Kent and Dodge and AlmaMater and Buell and EarlHall and
    StPaulChapel and Avery and OldComputerCenter and
    Chandler&Havemeyer and Schermerhorn and Uris near LowLibrary
Lewisohn and EarlHall and Chandler&Havemeyer near Mathematics
Philosophy and StPaulChapel and Avery and Schermerhorn near
    Fayerweather
Buell and StPaulChapel and LowLibrary and Fayerweather and
    Schermerhorn and Uris near Avery
LowLibrary and Chandler&Havemeyer and Uris near OldComputerCenter
LowLibrary and Mathematics and OldComputerCenter and Uris and
    NorthwestCorner near Chandler&Havemeyer
LowLibrary and Fayerweather and Avery and Uris and
    Mudd&EngTerrace&Fairchild&CS and SchapiroCEPSR near
    Schermerhorn
LowLibrary and Avery and OldComputerCenter and Chandler&Havemeyer
    and Schermerhorn and Mudd&EngTerrace&Fairchild&CS and
    SchapiroCEPSR and Pupin near Uris
Chandler&Havemeyer and Pupin near NorthwestCorner
Schermerhorn and Uris and SchapiroCEPSR near
    Mudd&EngTerrace&Fairchild&CS
Schermerhorn and Uris and Mudd&EngTerrace&Fairchild&CS and Pupin
    near SchapiroCEPSR
Uris and NorthwestCorner and SchapiroCEPSR near Pupin
```

4 Path generation

To generate a path, there were a few steps that had to be taken. First, all the relationship matrices from step 3 were combined into one matrix containing all the buildings that each building was related to. Iterating through this matrix, if a building had a relation, each building would be turned into a Vertex, an edge would be created between them, and this edge would be added to a graph of all the buildings. If a vertex already existed for a building, it would not be created, and edges were not bi-directional, meaning if A goes to B, B does not necessarily go to A. In order to determine the edge weights of this graph, the invisibility between two buildings was calculated. Essentially, the percent similarity between the combined what and where matrix were calculated, and this similarity became the cost of traveling from one node to another upon the graph.

After putting each building and all of its possible relations onto the graph, it was possible to run Dijkstra's algorithm in order to find the path of least confusability, rather than just the literal shortest path.

With the shortest path created, all that remained was to print out literal descriptions of the path and the source and target. Knowing the possible values within the list of what and where attributes, it was simple to filter out unnecessary information through a sequence of if-else statements. For example, if a building was not on any border it was unnecessary to include this information, but if it was on both the northern and western borders the building could be described as being on the north west corner of the map. Following is an example of a shape description for Uris:

```
the large symmetric complicatedRectangle building  that is
orientedNorth2South.
```

And following is an example of a shape description for the Hamilton, Hartley, Wallach, and John Jay Complex:

```
the not symmetric large CShaped building  that is
orientedNorth2South in the South-east corner .
```

A path consists of a direction from one location to the next location, including descriptions of each intermediate location. Below is the output when attempting to find the path between Pupin and the Northwest Corner building.

```
Begin at  Pupin , which is the  medium  symmetric complicatedRectangle
building  that is orientedEast2West on the northern border .
End at  NorthwestCorner , which is the  medium  symmetric rectangular
building  that is orientedNorth2South in the North-west corner .
Head  west to the medium  symmetric rectangular building  that is
orientedNorth2South in the North-west corner .
You have arrived at your destination
```

This path only has one step, since the Northwest Corner Building is directly west of Pupin. To account for confusability, we will demonstrate that the two most similar buildings, Mathematics and Lewisohn, are viewed different by the program. Below is the output from Mathematics to Low Library:

```
Begin at  Mathematics , which is the  medium  symmetric IShaped building
that is orientedNorth2South on the western border  in the northern half .
End at  LowLibrary , which is the  large  symmetric complicatedSquare
building  that has no single orientation .
Head  south to the small  symmetric complicatedRectangle building  that is
orientedEast2West.
Head  east to the large  symmetric complicatedSquare building  that has
no single orientation .
You have arrived at your destination
```

Now, below is the output from Lewisohn to Low Library:

```
Begin at  Lewisohn , which is the  medium  symmetric IShaped building
that is orientedNorth2South on the western border .
End at  LowLibrary , which is the  large  symmetric
complicatedSquare building  that is noSingleOrientation.
Head  north to the small  symmetric complicatedRectangle
building that is orientedEast2West.
Head  east to the large  symmetric complicatedSquare building
that has no single orientation.
You have arrived at your destination
```

Though these two are incredibly similar, it is impossible to confuse the two for two reasons. One, Lewisohn is south of Mathematics and has its own relationship to both Mathematics and Earl, not allowing them to be confused. The directions to earl differentiate the two, and the descriptions of Lewisohn and Mathematics differentiate the two if someone does not already know the difference. Mathematics is labeled as being on the North half of campus, while Lewisohn is not, since it is vertically centered on campus.

Now, an example of a longer path where the target cannot be reached with full certainty is the path from Mathematics to Philosophy, as can be seen below:

```
Begin at  Mathematics , which is the  medium  symmetric IShaped
building  that is orientedNorth2South on the western border  in
the northern half .
End at  Philosophy , which is the  medium  symmetric IShaped
building  that is orientedNorth2South on the eastern border.
Head  south to the small  symmetric complicatedRectangle
building  that is orientedEast2West.
Head to the medium  symmetric IShaped building  that is
orientedNorth2South on the western border .
Head  south to the large  symmetric rectangular building
that is orientedEast2West on the western border in the southern
half .
Head  north to the medium  symmetric IShaped building  that
is orientedNorth2South on the eastern border.
You have arrived at your destination
```

There are several buildings in between Mathematics and Pupin, so the route is not as simple as one who knows the area can tell it to be. However, seeing as Philosophy is visible from College Walk according to the campus map, this directional path still technically works.

Conclusion

By selecting a source and a target, the program was able to describe both buildings in understandable English, while also producing a descriptive path from source to target without relying on building names to provide a direction. Though not necessarily the paths that would be taken by one who knows their way around campus, the paths are descriptive enough to help one find their way around, and short enough so that they are not dragged to every building around campus before they find the building they are looking for. These descriptions were reasonably effective, as I forgot where Schermerhorn was on the map and the description given of the building helped me locate it. Though choosing not to use near in the directions, near was rather important in narrowing paths and selecting edges between two locations.