COMS W4735
Visual Interfaces to Computers
Assignment #3
Description of Visual Relations

Sabina Smajlaj (ss3912)

Table of Contents

[Part 1: The “What”]
[Part 2: The “Where”]
[Part 3: Source and Target Descriptions]
[Part 4: Creativity- Path Generation]

Part 1: The "What"

The main function of the first step was the describe each building using its location and shape vocabulary. We had 27 different buildings, so to be able to uniquely identify each building, that meant we needed ceiling(log27) bits of information for each one. This meant we had to have at least 5 bits of description information. However, since multiple buildings could possibly have the same descriptions, having more than five makes it easier to identify each building.

Hence I chose 16 different descriptions based on location and shape to identify the buildings. These were chosen based on unique identifiers for each buildings (ie. being L-shaped or I-shaped or smallest or largest) and unique identifiers for their locations (ie. being EasternMost or SouthernMost). These descriptions are as follows (in the form of functions):

Shape	Location
isSmall() isMedium() isLarge() isSmallest() isLargest() isRectangle() isSquare() isNonRectangle() isIShaped() isLShaped()	isLocatedCentrally() isLocatedOnBorder() isNorthernMost() isSouthernMost() isEasternMost() isWesternMost()

In order to compare the sizes for each building and determine what constituted a "small" or "large" building, I decided comparing area would be a good metric because it can be easily computed and is a standard unit of measurement. I began by first taking an average of the space of the image. Since the image was 275x495, this meant a possible 136125 pixels. Since about one third of that is black background, we can consider 90700 of the pixels to be occupied by buildings.

This meant that for the 27 buildings, each building on average took up an area 3360 pixels. Hence anything smaller than 2000 pixels could be considered "small", anything above 5000 can be considered “large”, and anything in between can be considered “medium.”

To get a more accurate description, I calculated the area of each building from the given image and made a graph of the different areas. From this, I discovered that about half of the map was actually black background, instead of one third. The values for the areas were: 1640, 1435, 5831, 5368, 5753, 3911, 3613, 322, 1164, 1182, 1191, 3898, 1087, 759, 1307, 1085, 340, 225, 1590, 1470, 4950, 2615, 5855, 920, 2240, 5282, 2240. The mean and standard deviation for the set were 2491 and 1909, respectively.

I then placed the areas on a bell curve and decided that anything below one standard deviation from the middle was “small”, and anything above one standard deviation from the middle was “large.” Hence, anything below 582 was small, anything above 4400 was large, and anything in between 582 and 4400 was medium.

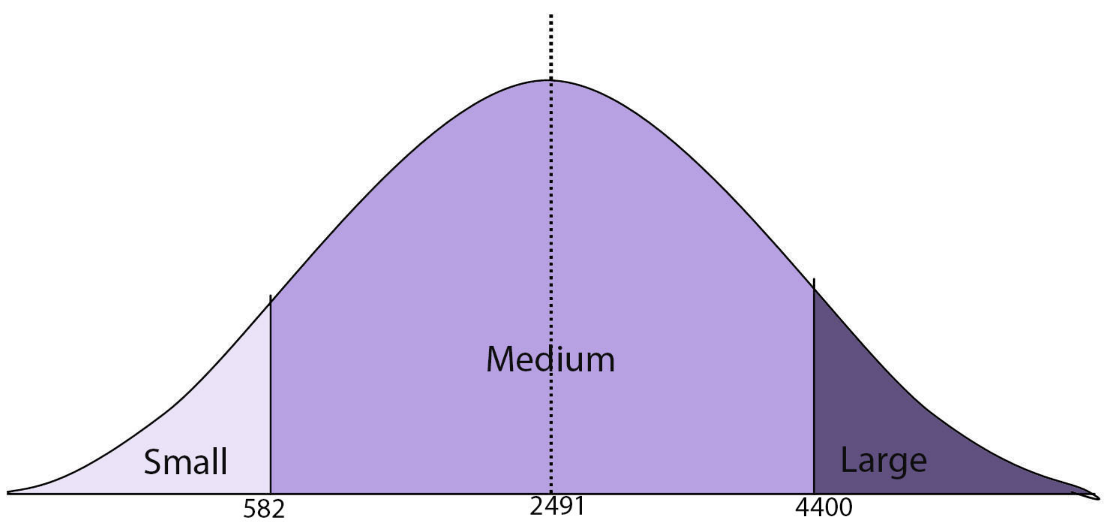


Figure #: A bell curve showing where the Small, Medium, and Large values fell.

The smallest and largest measures were also based on area. I kept track of the area of each building, and then compared all of them to determine the smallest and largest building on the image.

To determine whether each building was a Rectangle, Square, a NonRectangle, I-Shaped, or L-Shaped, I decided to use a method suggested by Professor Kender which was shape matching. Specifically within shape matching, template matching. I chose this method because it would be the quickest and most efficient. In addition, OpenCV supported template matching so I could rely on their libraries for accuracy. I created a sub image of each building and then had an image of each of the shapes (square, rectangle, I-shape, L-shape) that I used the OpenCV method matchTemplate() to determine whether the building matched any of the templates.

To determine the location of the building, I first began by finding all of the edges on the given image using the Canny() method within OpenCV. This is shown in Figure # below:

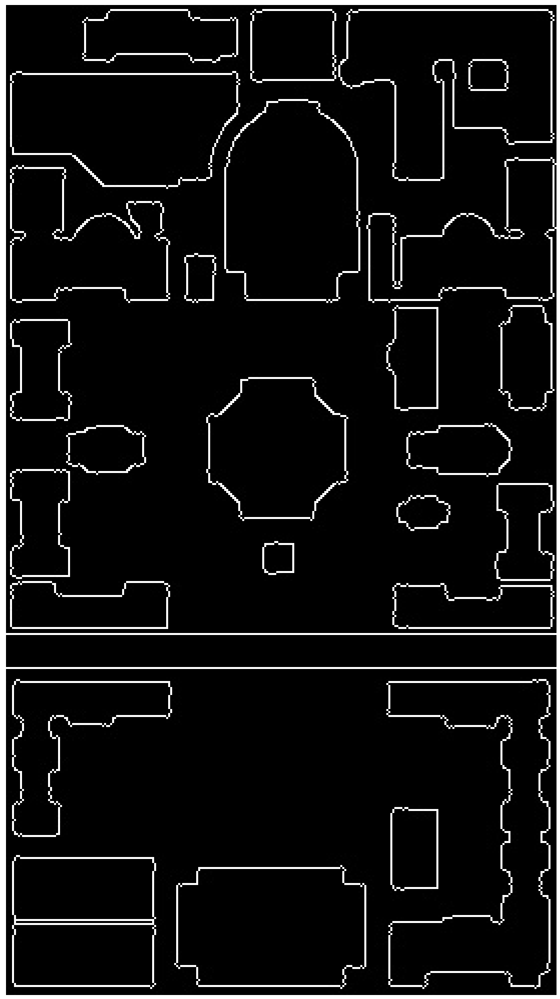


Figure #: The edges of the buildings.

Then from the edges, I found each contour on the grid using the findContours() method. This represented each building and is shown below:

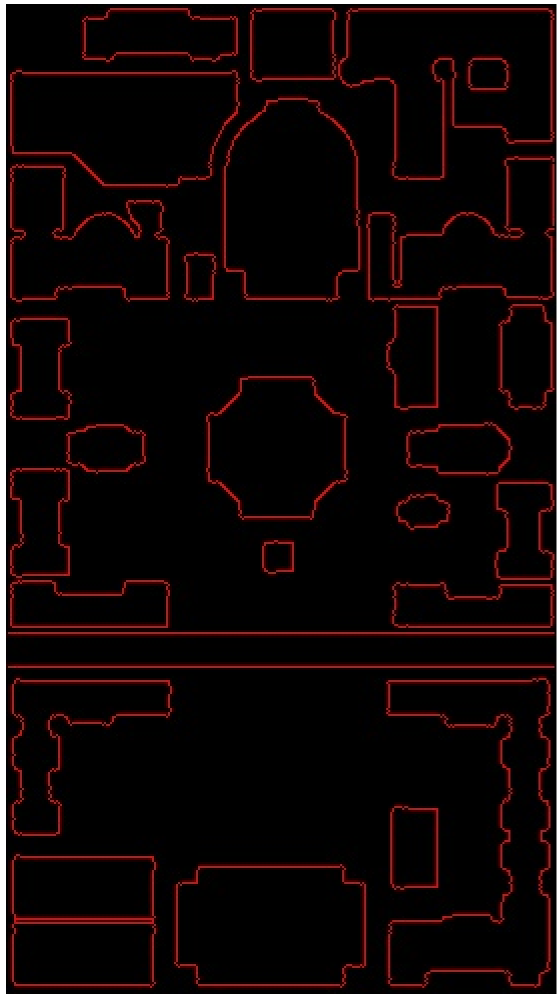
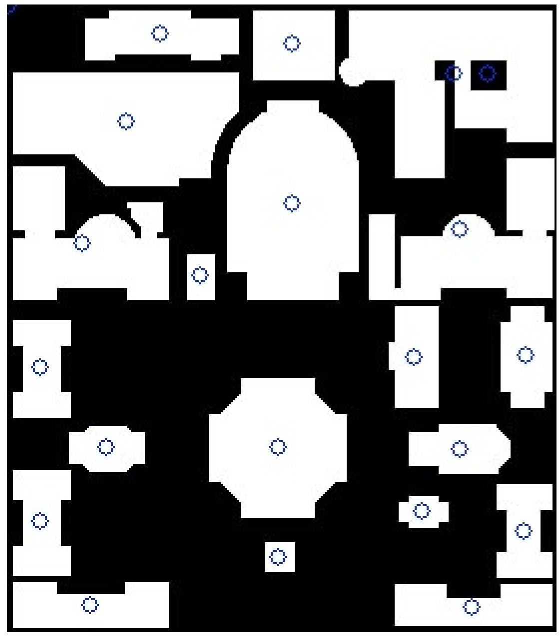


Figure #: The contours of the buildings.

From the contours, I was able to get the moments and then the center of mass of each building.



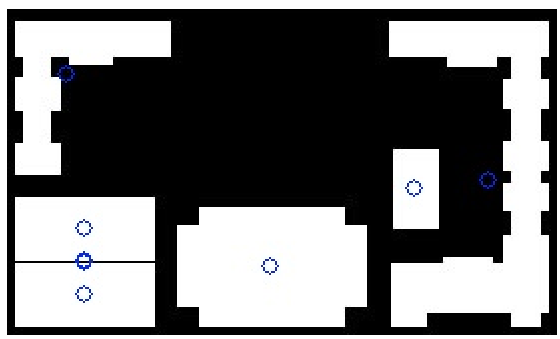


Figure #: The moments of each building.

However, many buildings had borders that were perfectly aligned at the edges of each direction, ie. we can see in Figure # below that Carman, Butler, and John Jay all have borders that are aligned at the bottom of the image. Hence, merely looking at the southernmost pixels of each image would not give us any unique results for southernmost, northernmost, easternmost, or westernmost; each of these would have at least three buildings that satisfy the requirement. However, looking at the values for the center of mass would give us more interesting and accurate (and unique) results.

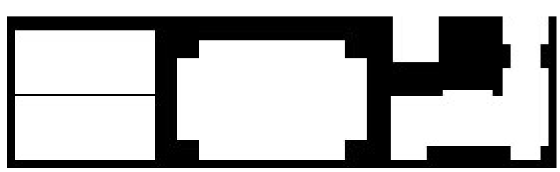


Figure #: The southern borders of Carman, Butler, and John Jay/etc are all aligned.

Even though we cannot use the border pixels of each building to determine the southernmost, etc., we can use them to determine whether each building is located centrally or on the border. From the image below in Figure #, we can see that buildings located on the border will contain pixels all in the same x or y location. The buildings in the example below are all located on the Western border, hence they will all contain pixels that fall on (x,25). This will hold true for any buildings on the border for the other three compass directions. We can also see from the image below that Earl would not contain a pixel that falls into this coordinate, hence it would not be in the border. And as our human eyes can verify, Earl is not a border building.

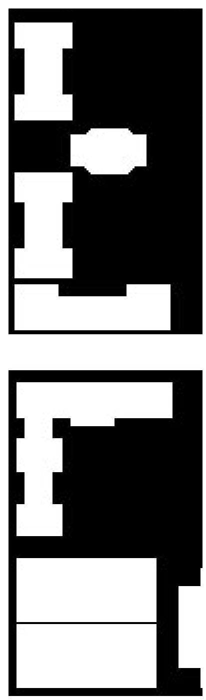


Figure #: All of these buildings are located on the Western border.

My results are as follows:

```
Building Number: 1
Building Name: "Pupin"
Center of Mass Coors: 76 ,14
It has area 1640
It is a medium building
It is located on the border
It is the northern most building.

Building Number: 2
Building Name: "Schapiro CEPSR"
Center of Mass Coors: 142 ,19
It has area 1435
It is a medium building
It is located on the border

Building Number: 3
Building Name: "Mudd, Engineering Terrace, Fairchild & Computer Science"
Center of Mass Coors: 223 ,54
It has area 5831
It is a large building
It is located on the border

Building Number: 4
Building Name: "Physical Fitness Center"
Center of Mass Coors: 59 ,58
It has area 5368
It is a large building
It is located on the border

Building Number: 5
Building Name: "Gymnasium & Uris"
Center of Mass Coors: 142 ,99
It has area 5753
It is a large building
It is located on the border

Building Number: 6
Building Name: "Schermerhorn"
Center of Mass Coors: 226 ,124
It has area 3911
It is a medium building
It is located on the border

Building Number: 7
Building Name: "Chandler & Havemeyer"
Center of Mass Coors: 37 ,119
It has area 3613
It is a medium building
It is located on the border

Building Number: 8
Building Name: "Computer Center"
Center of Mass Coors: 96 ,135
It has area 322
It is a small building
It is located centrally
```

```
Building Number: 9
Building Name: "Avery"
Center of Mass Coors: 203 ,176
It has area 1164
It is a medium building
It is located on the border

Building Number: 10
Building Name: "Fayerweather"
Center of Mass Coors: 259 ,175
It has area 1182
It is a medium building
It is located on the border
It is the eastern most building.

Building Number: 11
Building Name: "Mathematics"
Center of Mass Coors: 16 ,181
It has area 1191
It is a medium building
It is located on the border
It is the western most building
It is I-Shaped

Building Number: 12
Building Name: "Low Library"
Center of Mass Coors: 135 ,221
It has area 3898
It is a medium building
It is located on the border

Building Number: 13
Building Name: "St. Paul's Chapel"
Center of Mass Coors: 226 ,222
It has area 1087
It is a medium building
It is located on the border

Building Number: 14
Building Name: "Earl Hall"
Center of Mass Coors: 49 ,221
It has area 759
It is a medium building
It is located on the border

Building Number: 15
Building Name: "Lewisohn"
Center of Mass Coors: 16 ,258
It has area 1307
It is a medium building
It is located on the border
It is I-Shaped
```

```
Building Number: 16
Building Name: "Philosophy"
Center of Mass Coors: 258 ,263
It has area 1085
It is a medium building
It is located on the border
It is I-Shaped

Building Number: 17
Building Name: "Buell & Maison Francaise"
Center of Mass Coors: 136 ,276
It has area 340
It is a small building
It is located on the border

Building Number: 18
Building Name: "Alma Mater"
Center of Mass Coors: 136 ,322
It has area 225
It is a small building
It is the smallest building
It is located centrally
It is a square

Building Number: 19
Building Name: "Dodge"
Center of Mass Coors: 41 ,300
It has area 1590
It is a medium building
It is located on the border

Building Number: 20
Building Name: "Kent"
Center of Mass Coors: 232 ,300
It has area 1470
It is a medium building
It is located on the border

Building Number: 21
Building Name: "College Walk"
Center of Mass Coors: 137 ,322
It has area 4950
It is a large building
It is located on the border
It is a rectangle

Building Number: 22
Building Name: "Journalism & Furnald"
Center of Mass Coors: 29 ,364
It has area 2615
It is a medium building
It is located on the border
It is L-Shape
```

```
Building Number: 23
Building Name: "Hamilton, Hartley, Wallach & John Jay"
Center of Mass Coors: 240 ,417
It has area 5855
It is a large building
It is the largest building
It is located on the border

Building Number: 24
Building Name: "Lion's Court"
Center of Mass Coors: 203 ,421
It has area 920
It is a medium building
It is located on the border

Building Number: 25
Building Name: "Lerner Hall"
Center of Mass Coors: 38 ,441
It has area 2240
It is a medium building
It is located on the border
It is a rectangle

Building Number: 26
Building Name: "Butler Library"
Center of Mass Coors: 131 ,460
It has area 5282
It is a large building
It is located on the border

Building Number: 27
Building Name: "Carman"
Center of Mass Coors: 38 ,474
It has area 2240
It is a medium building
It is located on the border
It is the souther most building
It is a rectangle
```

The language was written just as sentences. Although it does not have the best sentence structure, having one sentence for each description made it easier to keep track of each description and make sure it made sense to human readers. rtr

Part 2: The "Where"

For step 2, we had to describe the "WHERE" of each building by comparing their locations to the buildings around them. This meant trying to decipher if one building was north of another building, south of another building, east of another building, west of another building, or just near another building.

The first four attributes (north, south, east, west) may depend only on the location of each building, and not on their size. While this gives an idea of their locations and how they compare to each other, it does not provide the most accurate results. For example, as we can see in Figure # below, when we're looking at the Dodge Fitness Center and Havemayer/Chandler, their centers of mass would suggest that the Dodge Fitness Center is East of Havemayer/Chandler. However, looking at them with the human eye, we can see that they are at approximately the same location. Hence taking size and shape into account would predict much more accurate results.

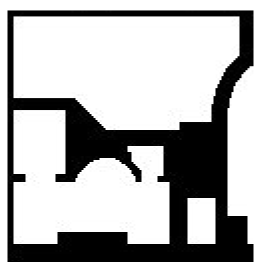


Figure #: The building above is Dodge Fitness Center and the building below is Havemayer/Chandler.

Nearness has to depend on the size of each building. All five attributes are also only true in the given direction, ie. if Building1 was near Building2, that did not necessarily mean that Building2 would be near Biulding1.

To calculate the compass directions for two buildings (a Source and a Target), I started by using the center of mass of each building that was calculated in Part 1. Since the center of mass provides an intuitive and practical measure for where the "middle" of the image should be, we can use it to compare the relative placements of two buildings. If the pixel coordinates for the center of mass of the Source are (100, 150) and the coordinates for the Target are (50, 75), then we can judge that the Source is South of the Target and East of the Target.

However, as I mentioned above, this does not provide accurate enough comparisons on its own. Hence, I then had to take size and shape into account. The larger a building was in area, the further its dimensions could be from another building and they could still satisfy the relationship.

The next step involves reducing the relation descriptions for each building. For each building, we compared it to each of the other 26 buildings using the 5 binary relations described (North, South, East West, Near). This means that we will have 27*26*5= 3510 possible comparison for each building (we're multiplying by 26 because we don't need to compare the buildings to themselves).

However, we can reduce this number of relations based on transitivity. Transitivity in computer vision works much the same way as in mathematics: if A is related to B, and C is related to C, then A is related to C. In our program, this translates to: if Building1 is East of Building2, and Building 2 is East of Building3, then Building1 must be East of Building2.

The way I defined the Transitivity Law meant that transitivity applies to North, South, East, and West. Even though Abella argued that transitivity did not apply to North, I am defining anything with a center of mass y-coordinate that is smaller (and hence higher in the image since pixel counts start at (0,0) at the top-left corner) to be above a center of mass y-coordinate that is larger. In the example in Figure # below taken from the Abella thesis, (a) clearly has a smaller y-coordinate than both (b) and (c). Although (a) is not directly above (c), my definition of above ("North") still defines this relationship as holding true. (a) is both North of and East of (c).

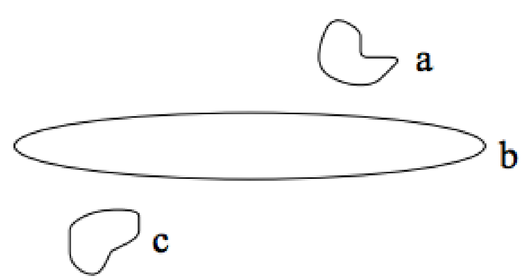


Figure #: Shapes indicating above placement.

Hence transitivity would mean that since we know that (a) is above (b), and (b) is above (c), then we can infer that (a) is above (c) and hence don't necessarily need to state this relationship. So we can reduce this relationship to get closer to our minimum amount of relations.

However, as stated in the Abella Thesis, transitivity does not apply to Near. My reasoning lies along the same lines as described in Abella. Nearness can be defined in a multitude of ways, but the most simple and practical would be to define it as relating to the objects immediately surrounding a Source. For our program, this would mean the buildings that are immediately surrounding our Source building. In Figure X below, we can see that Earl is “near” Low Library. We can also see that Low Library is “near” St. Paul’s Chapel. However, this does not infer that Earl is “near” St. Paul’s Chapel. In fact, Earl and St. Paul’s Chapel are not near each other at all; they are on opposite sides of the campus.

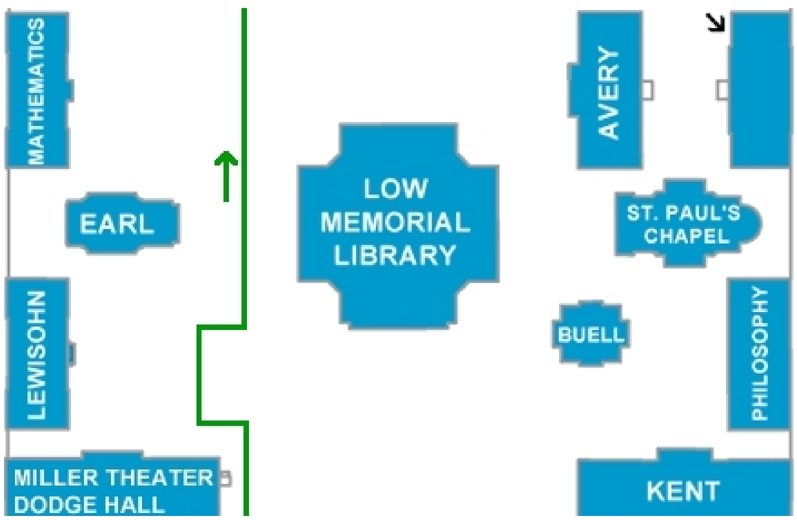


Figure #: Map of the central campus buildings.

One way to determine which buildings are “near” our Source building is to mimic the human experience and find out which ones are in the immediate surroundings. This is done by forming a bounding box for the Source building that matches its shape. Then, we can expand the bounding box (keeping the center at the same location) by a certain percentage, and any building that is intersecting with that bounding box can be considered to be “near” it. The expansion would occur uniformly on all sides.

The results are as follows:

```
Building 1 is south of 0
Building 1 is near 2
Building 1 is north of 2
Building 1 is near 4
Building 1 is east of 4
Building 1 is near 5

Building 2 is near 1
Building 2 is south of 1
Building 2 is north of 3
Building 2 is near 5
Building 2 is east of 21

Building 3 is south of 2
Building 3 is north of 4
Building 3 is near 6
Building 3 is east of 9
Building 3 is east of 24

Building 4 is near 0
Building 4 is near 1
Building 4 is south of 3
Building 4 is north of 5
Building 4 is near 7
Building 4 is near 8
Building 4 is east of 14

Building 5 is near 1
Building 5 is near 2
Building 5 is south of 4
Building 5 is north of 7
Building 5 is near 8
Building 5 is near 9
Building 5 is east of 21

Building 6 is near 3
Building 6 is east of 3
Building 6 is south of 7
Building 6 is north of 8
Building 6 is near 9
Building 6 is near 10
Building 6 is near 13

Building 7 is near 4
Building 7 is south of 5
Building 7 is north of 6
Building 7 is near 8
Building 7 is near 11
Building 7 is east of 22

Building 8 is east of 1
Building 8 is near 4
Building 8 is near 5
Building 8 is south of 6
Building 8 is near 7
Building 8 is north of 10
Building 8 is near 12
Building 8 is near 14
```

```
Building 9 is east of 2
Building 9 is near 5
Building 9 is east of 5
Building 9 is near 6
Building 9 is near 10
Building 9 is south of 10
Building 9 is north of 11
Building 9 is near 12
Building 9 is near 13
Building 9 is near 16

Building 10 is near 6
Building 10 is south of 8
Building 10 is near 9
Building 10 is north of 9
Building 10 is near 13
Building 10 is near 16
Building 10 is east of 16

Building 11 is east of 0
Building 11 is near 7
Building 11 is south of 9
Building 11 is north of 12
Building 11 is near 14
Building 11 is north of 14
Building 11 is near 15

Building 12 is near 8
Building 12 is near 9
Building 12 is south of 11
Building 12 is north of 13
Building 12 is near 17
Building 12 is east of 26

Building 13 is east of 3
Building 13 is near 6
Building 13 is near 9
Building 13 is near 10
Building 13 is south of 12
Building 13 is south of 14
Building 13 is north of 15
Building 13 is near 16
Building 13 is near 20

Building 14 is near 8
Building 14 is near 11
Building 14 is south of 11
Building 14 is north of 13
Building 14 is near 15
Building 14 is near 19
Building 14 is east of 19

Building 15 is east of 0
Building 15 is near 11
Building 15 is south of 13
Building 15 is near 14
Building 15 is north of 16
Building 15 is near 19
```

```
Building 16 is near 9
Building 16 is near 10
Building 16 is near 13
Building 16 is south of 15
Building 16 is north of 17
Building 16 is near 20
Building 16 is east of 23

Building 17 is near 12
Building 17 is east of 12
Building 17 is south of 16
Building 17 is near 18
Building 17 is north of 19
Building 17 is north of 20
Building 17 is near 21

Building 18 is east of 12
Building 18 is near 17
Building 18 is south of 19
Building 18 is south of 20
Building 18 is near 21
Building 18 is north of 22
Building 18 is near 24

Building 19 is near 14
Building 19 is near 15
Building 19 is south of 17
Building 19 is north of 18
Building 19 is north of 21
Building 19 is near 22
Building 19 is east of 25
Building 19 is east of 27

Building 20 is east of 6
Building 20 is near 13
Building 20 is east of 13
Building 20 is near 16
Building 20 is south of 17
Building 20 is north of 18
Building 20 is north of 21

Building 21 is near 17
Building 21 is east of 17
Building 21 is near 18
Building 21 is east of 18
Building 21 is south of 19
Building 21 is south of 20
Building 21 is north of 22
Building 21 is near 24

Building 22 is east of 11
Building 22 is east of 15
Building 22 is south of 18
Building 22 is near 19
Building 22 is south of 21
Building 22 is north of 23
Building 22 is near 25
```

```
Building 23 is east of 20
Building 23 is south of 22
Building 23 is near 24
Building 23 is north of 24

Building 24 is east of 2
Building 24 is east of 5
Building 24 is near 18
Building 24 is near 21
Building 24 is near 23
Building 24 is south of 23
Building 24 is north of 25
Building 24 is near 26

Building 25 is east of 7
Building 25 is near 22
Building 25 is south of 24
Building 25 is north of 26
Building 25 is near 27

Building 26 is east of 8
Building 26 is near 24
Building 26 is south of 25
Building 26 is north of 27

Building 27 is east of 7
Building 27 is near 25
Building 27 is south of 26
```

Part 3: Source and Target Descriptions

In step 3, we want to create the infrastructure for our system. We want to be able, for any (x,y) location on the image, to describe that location using the definitions and relationships we developed in Part 1 and Part 2.

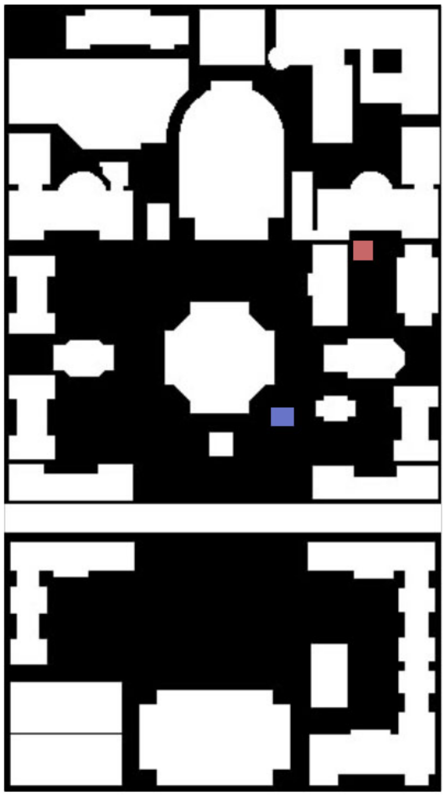
This (x,y) location is treated as a new one-pixel building that has been added to the map and can be part of a building or part of the black background. If it is part of a building, that we can consider it primarily as being "in" that building. If it is part of the black background, then we want to consider that location with respect to the building that it is "nearest" to.

Since this new one-pixel building is a new location, we want to use the functions we developed in Part 2 to describe its "where" (there is no "what" for this since it is just one pixel) in relation to existing buildings. This holds for both the Source and the Target that the user clicks on, where the Source is the first recorded click and the Target is the second recorded click. Since I had written my "where" descriptions based on the center of mass of the buildings, I just treated the new pixel as it's own center of mass, since it technically is.

To get these descriptions, I compared the new pixel with each of the 27 buildings using the 5 attributes. I then reduced these descriptions again using the methods defined in Part 2, to get a unique description of the location of the pixel.

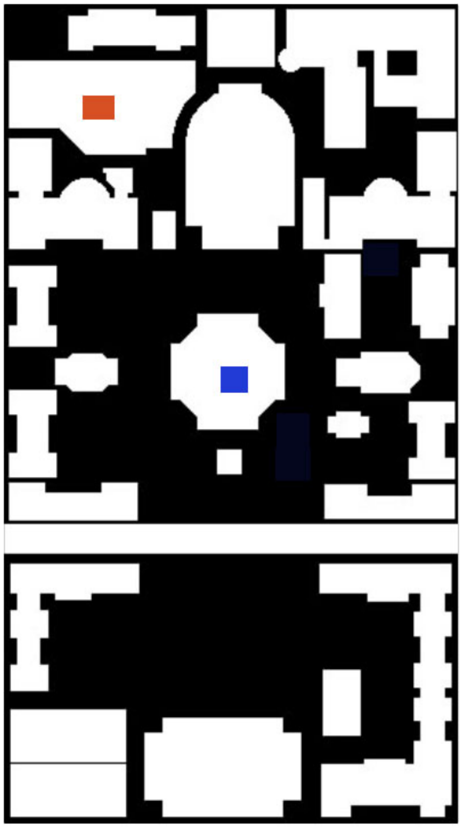
However, since there are 136125 possible pixels on the image, there will be a multitude of pixels with the same descriptions; these pixels will be located near each other. Hence, each description will likely apply to a "class" of pixels, which we will refer to as a "cloud". Hence, for each location on the grid, there will be a "cloud" of pixels surrounding it that share the same description as that location.

Note: I couldn't get my macbook to print a screenshot that included the mouse, so I can't show you the mouse clicks. I'll just place squares on the screenshots of the images, red for first click and blue for second click.



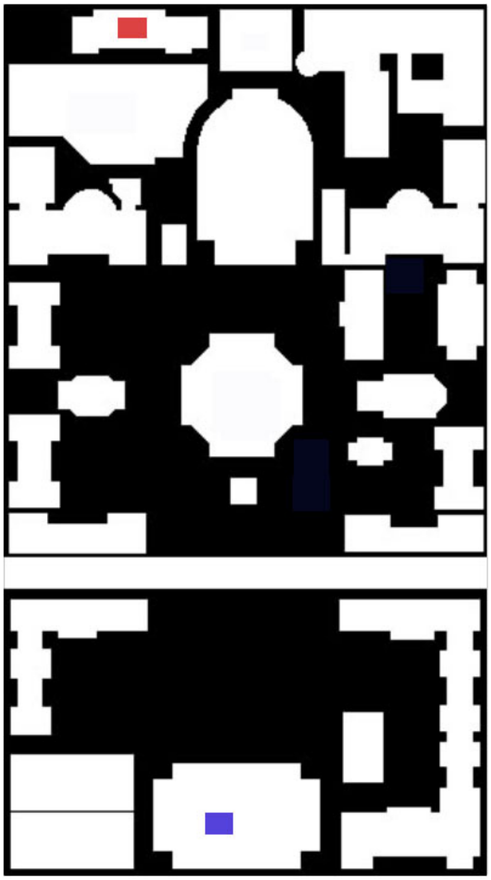
```
Building 28 is near Building 3
Building 28 is near Building 6
Building 28 is south of Building 8
Building 28 is near Building 9
Building 28 is east of Building 9
Building 28 is near Building 10
Building 28 is north of Building 10
Building 28 is near Building 13
Building 28 is east of Building 24
```

```
Building 29 is east of Building 2
Building 29 is east of Building 5
Building 29 is near Building 9
Building 29 is near Building 12
Building 29 is near Building 13
Building 29 is south of Building 16
Building 29 is near Building 17
Building 29 is north of Building 17
Building 29 is near Building 18
Building 29 is near Building 20
Building 29 is near Building 21
```



```
Building 30 is near Building 1
Building 30 is near Building 4
Building 30 is south of Building 4
Building 30 is north of Building 5
Building 30 is near Building 7
Building 30 is near Building 8
Building 30 is east of Building 14
```

```
Building 31 is east of Building 2
Building 31 is east of Building 5
Building 31 is near Building 8
Building 31 is near Building 9
Building 31 is near Building 12
Building 31 is south of Building 13
Building 31 is north of Building 15
Building 31 is near Building 17
Building 31 is near Building 18
Building 31 is near Building 21
```



```
Building 32 is near Building 1
Building 32 is north of Building 1
Building 32 is near Building 2
Building 32 is near Building 4
Building 32 is east of Building 4
Building 32 is near Building 5
```

```
Building 33 is east of Building 8
Building 33 is near Building 25
Building 33 is near Building 26
Building 33 is south of Building 26
Building 33 is near Building 27
Building 33 is north of Building 27
```

Part 4: Creativity- Path Generation

In step 4, we want to describe a path from the Source (S) to the Target (T). The Source and Target can be anywhere in the map, similar to Part 3. The path from Source to Target should be dependent on the intermediate buildings between them; this path should consist of the shortest path of buildings from S to T.

The best way to get this shortest path is to run a "search" at each step for a "good" set of directions.

Since I couldn't get Part 40 to work, I'll describe what I would have done.

The next step involved having friends test your path generation. First we generate eight paths for eight different (S, T) pairs. Then we print out these directions and have each of our two friends test these. The same user interface from Part 3 was used. Each friend started at the Source and clicked on the map for each step of the directions. To keep track of each step and where the user clicked, my program generated a new image on each click that highlighted (via a rectangle) the area of the map that was clicked.

We also wanted to have an option in case the user did not understand the directions. This was in the form of an "I'm lost" click. For my program, I positioned this click to be at the Top-Left corner as seen below in Figure #, because there are no buildings in that location which would confuse the program.

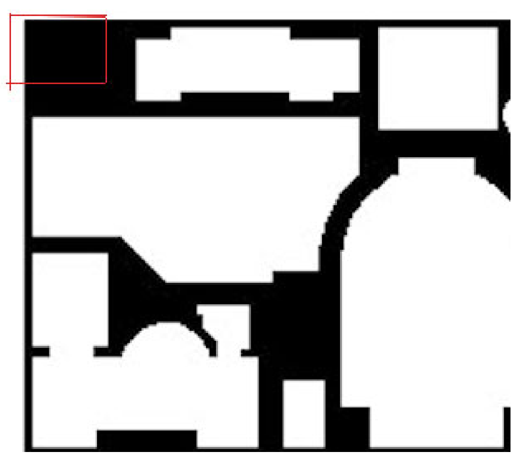


Figure #: The position of the "I'm Lost" click is shown in red at the Top-left of the map.