GEO877 Aiello's Group - Key Algorithms Applied:

1) Spatial Indexing
2) Buffering street segments
3) Polygon point intersection via ray method

Spatial indexing

Since the main operation of our project is an intersection between more than 150'000 points (in our case: Flickr photos) and over 290'000 polygons (in our case: street segment buffers), a brute force approach is computationally extremely intensive.

To facilitate efficient spatial analysis, we created a spatial indexing system for New York City's street data. We began by defining a bounding box around the street network, which we divided into a grid with cells measuring 2,000 meters on each side. This resulted in a 24 x 24 grid overlaying the street network (Fig. 2). Utilizing Haversine distances allowed us to maintain accuracy without converting to a metric coordinate system, as it accounts for the Earth's curvature. However, since the Haversine distance is only an approximation, the corner points of the grids don't exactly lie on each other but are a bit shifted to the right (Fig. 1). Nevertheless, the whole bounding box area is covered without any spatial gaps which is crucial to store all points.

The grid cells were populated with geotagged Flickr photos by comparing the x and y coordinates of the photos with those of the grid cells. This process enabled the creation of a data frame that stored the grid cell number, its bounding box, and the corresponding photos. For more efficient processing, this data frame was later converted into a dictionary since we found queries using dictionaries to be much faster than such using Pandas data frames. The purpose of this spatial indexing was to accelerate the upcoming point-in-polygon intersection, a critical step in associating photos with specific street segments. The Coordinate Reference System (CRS) used for this task was WGS84, ensuring consistent and accurate spatial calculations. This indexing system formed the foundation for our subsequent spatial analysis and the development of an urban sound map for New York City.

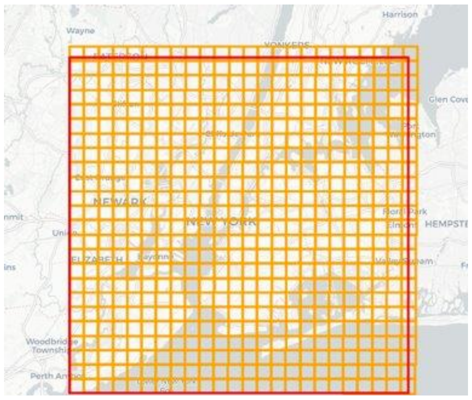
Figure 2: Spatial index (orange) and street network bounding box (red).

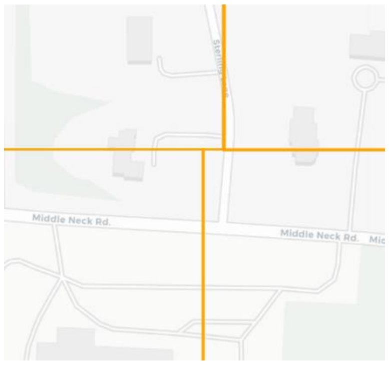

Figure 1: The slight spatial shift due to the approximative matter of the Haversine distance.

Buffering segments in WGS84

To assign photos and their tags to specific street segments, we had to create buffers around these segments. To avoid the complexities of reprojecting data into a metric coordinate system, we performed the buffering in the WGS84 coordinate system using Haversine distances again. This approach ensures that the Earth's curvature is accurately accounted for and allows to bypass computationally expensive CRS reprojection into a metric system. However, it also introduces additional complexity since metric vectors cannot be used. Instead, we had to calculate the coordinates of the buffer's corner points using the bearing (or azimuth) angle in combination with the Haversine distance. We set the buffer width to 20 meters, consistent with the methodology in Aiello's paper.

First, we calculated the bearing of the segment's start and its end point in both directions using a complex formula found on this page. After verifying this approach using interactive and zoomable folium maps, we used these two bearings and the set buffer width of 20 meters to get two interim points to the left and to the right of the segment. Therefore, we defined a function which took a start point, the bearing angle of the desired destination point and the distance (in this case: the buffer width of 20 m) to retrieve the destination point in WGS84. From the interim points, we then again took the initial bearings and defined their perpendicular bearings to finally get the four corner points of the buffer. Eventually, each street segment was thus surrounded by a rectangular buffer, ensuring accurate spatial relationships between the segments and the geotagged photos. An example in the accompanying figure illustrates a street segment shown in blue and its corresponding buffer in orange (Fig. 3). This buffering process allowed us to effectively assign photos and their tags to the relevant street segments, forming a critical step in our analysis. The method ensured that we could handle the data efficiently and keep all spatial information in the original coordinate system of the street data (which is WGS84).

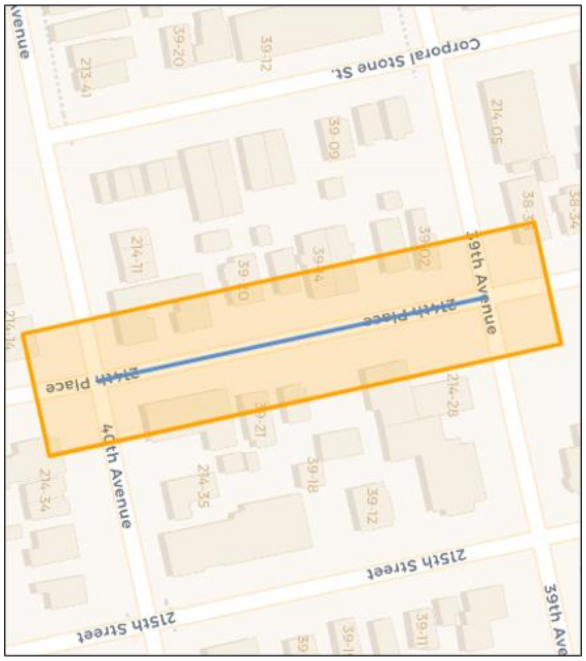

Figure 3: A street segment (blue) and its corresponding buffer (orange).

Polygon point intersection via ray method

To determine if photos inside the segment buffers, we employed a point-in-polygon method. Here, we made use of the spatial index. First, we added bounding boxes to all street segment buffers and then performed an intersection between all these buffer bounding boxes and all grid cells. This has the efficiency of n^2 which is usually not desirable but can't be solved differently. Here, the creation of buffer bounding boxes helps to make the intersection faster compared to using the buffers itself since bounding box to bounding box intersection simply compares the latitude and longitude values of the two polygons. Furthermore, we again used the trick of querying information via dictionaries which was found to be about 50 times quicker for this bounding box to bounding box intersection. Hence, this inefficient (n^2) intersection process is really quick. Eventually, each segment buffer got a new column with one or more corresponding grid cell(s) whereas each grid cell was already populated with the respective photos relevant for the segment buffer (see page 1).

After, each relevant photo point within these grid cells was then analyzed in relation to the corresponding street segment buffer. We used the "even-odd" method for this analysis, which involves drawing a horizontal ray in the positive x direction from the point. By counting the number of times this ray intersects the edges of the segment buffer polygon, we could determine whether the point lies inside (odd number of intersections) or outside (even number of intersections) the polygon. For example, if a photo point intersects the buffer an odd number of times, it is inside the buffer and thus assigned to the corresponding street segment. This "even-odd" method is both efficient and effective for spatial analysis, ensuring precise association of geotagged photos (and their tags) with street segments. Regarding efficiency, we further tested the "always-left" method which is only possible if the polygons are rectangles which is the case for our street segment buffers. This method makes use of the determinant and the relative location of a point regarding an edge of the rectangle. However, using "timeit" found this approach to be less efficient. Hence, the "even-odd" method was applied to all street segments, allowing us to accurately assess and assign photos to their respective segment buffers.

Figure 4 illustrates the process, showing a street segment in blue, its buffer in orange, and an intersecting point in orange. By leveraging the intersection of the segment buffer bounding box and the grid cells, we ensured that only relevant points were considered, optimizing the computational efficiency of our analysis. This comprehensive method allowed us to robustly analyze and map the spatial distribution of urban sounds in New York City.

Generally, the efficiency of this final intersection is mainly dependent on the resolution of the spatial index as larger grid cells are populated with more photos leading to a more intensive photo in buffer intersection but a faster segment buffer in grid cell intersection (and the other way around).

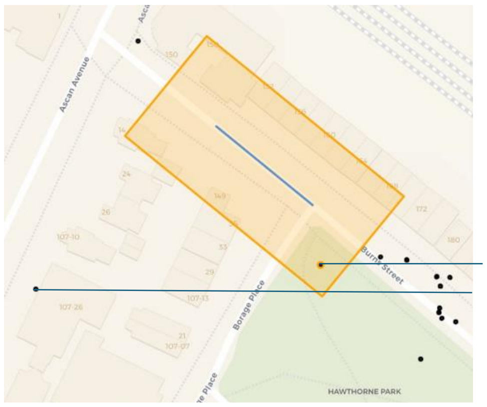

Figure 4: The "even odd" method which counts the number of polygon edges that intersect with a horizontal ray (dark blue). If the number is even, the point is outside the polygon, if it is odd, it's inside.