**Trajectory Data Extraction and Traffic Performance Analysis Using a Stationary Drone in SUMO**

Site Description:

The Pratunam intersection in Bangkok is a major, chronically congested signalized junction. It connects key commercial zones like Platinum Mall and Siam Square, handling extremely high volumes of mixed traffic, including cars, motorcycles, and buses, making it a critical focus for traffic management studies.

**Area coverage by the Drone image**

**Given Parameters**

| **Parameter** | **Value** | **Meaning** |
| --- | --- | --- |
| IMG\_WIDTH | 1920 px | Image width (horizontal) |
| IMG\_HEIGHT | 1080 px | Image height (vertical) |
| SCALE | 5 px/m | Scale = 5 pixels per meter |

**Step 1: Compute the real-world width and height**

Since SCALE = 5 px per meter:

So, each frame covers **384 m × 216 m** of real-world area.

**Step 2: Determine coordinate coverage around the drone center**

The camera is centered at (CAM\_X, CAM\_Y) = (288.84, 187.33).  
Half of the width/height will extend in each direction:

* Half-width = 384 / 2 = **192 m**
* Half-height = 216 / 2 = **108 m**

So, the **covered area in world coordinates** is approximately:

| **Axis** | **Range (meters)** |
| --- | --- |
| X-axis | 288.84 − 192 → 288.84 + 192 = **[96.84, 480.84]** |
| Y-axis | 187.33 − 108 → 187.33 + 108 = **[79.33, 295.33]** |

**Final Answer:**

* **Frame coverage:** 384 m (X) × 216 m (Y)
* **World coordinate coverage:**
  + X range: **96.84 – 480.84 m**
  + Y range: **79.33 – 295.33 m**

**TraCI Script for Drone Based Vehicle Trajectory Data Collection at a Fixed Location**

* **Initialize Configuration**
  + Set SUMO config file paths
  + Define simulation step length (1.0 second)
  + Configure drone: position (X,Y), movement speed, view scale, image dimensions
  + Set output directories for CSV and frames
* **Setup Environment**
  + Create output directories for data and images
  + Start SUMO simulation using TraCI connection
* **Main Simulation Loop** (while vehicles exist)
  + Get current simulation time
  + Create blank drone view image
  + **For each vehicle in simulation:**
    - Extract position, speed, and lane ID
    - Write vehicle data to CSV (time, ID, position, speed, lane)
    - Convert world coordinates to pixel coordinates
    - Draw vehicle as rectangle on drone image
  + Save drone view frame as image file
  + Update drone position (if moving)
  + Advance simulation to next step
* **Cleanup**
  + Close TraCI connection

**Python code to extract vehicle trajectory from the drone footage**

* **Load Data**
  + Read vehicle trajectory data from Excel file into DataFrame
* **Initialize Plot**
  + Create figure with specified size for space-time diagram
* **Prepare Vehicle Data**
  + Extract all unique vehicle IDs from dataset
  + Generate color palette for distinguishing different vehicles
* **Process Each Vehicle**
  + **For each unique vehicle ID:**
    - Filter and sort data by time for current vehicle
    - **If vehicle has sufficient data points:**
      * Extract position coordinates (x, y) and timestamps
      * Calculate distance traveled between consecutive points
      * Compute cumulative distance from starting position
      * Plot time vs cumulative distance with unique color and markers
* **Finalize Visualization**
  + Add labels for axes and title
  + Enable grid for better readability
  + Display legend outside main plot area
  + Adjust layout and render the space-time diagram

**Python Code for Aggregated Vehicle Count, Travel Time, and Speed**

* **Function: calculate\_traffic\_metrics**
  + Convert time to integer seconds
  + Sort data by vehicle ID and time for chronological tracking
  + Create 60-second time intervals covering simulation duration
  + **For each time interval:**
    - Filter data for current time window
    - Calculate unique vehicle count
    - Compute average speed across all observations
    - **For each vehicle in interval:**
      * Calculate travel time as time between first and last appearance
      * Collect valid travel times (> 0 seconds)
    - Compute average travel time from collected values
    - Store metrics for current interval
* **Main Execution**
  + Load trajectory data from Excel file
  + Call metrics calculation function
  + Display formatted results table with:
    - Time intervals
    - Vehicle counts
    - Average travel times
    - Average speeds
* **Visualization**
  + Create three subplots for different metrics
  + Plot vehicle counts as bar chart
  + Plot average speeds as line graph with markers
  + Plot average travel times as line graph with markers
  + Save plot as image file
* **Output**
  + Save detailed metrics table to CSV file