1 Psuedocode

Algorithm 1 Congestion Zone Detection

procedure FindAndBuildCongestionZones(*trajectories*,
min_traj_duration,
direction,
light_thresh,
heavy_thresh)

    $l\_points, h\_points, l\_speeds, h\_speeds \leftarrow$ empty arrays

    for each trajectory **do**
        if $trajectory.direction \neq direction$ **then continue**
        end if
        if $trajectory.duration < min\_traj\_duration$ **then continue**
        end if

        $x \leftarrow trajectory.x\_positions$, $t \leftarrow trajectory.timestamps$
        $speed \leftarrow \Delta x / \Delta t \cdot direction \cdot 0.681818$ ▷ speed in each point

        for threshold in {light_thresh, heavy_thresh} **do**
            $mask \leftarrow (speed > 0) \wedge (speed \leq threshold)$
            Detect contiguous segments in *mask*
            for each segment $[start, end]$ **do**
                Append $(x[start], t[start]), (x[end], t[end])$ to points
                Append $speed[start], speed[end]$ to speeds
            end for
        end for
    end for

    $l\_polygons \leftarrow$ CreateCongestionZones$(l_points, l_speeds, l_params)$
    $h\_polygons \leftarrow$ CreateCongestionZones$(h_points, h_speeds, h_params)$

    return $(l\_polygons, h\_polygons)$

end procedure

Algorithm 2 Sub-algorithms

```
procedure CreateCongestionZones(
points, speeds, params)
    P ← BuildParallelograms(points, speeds)
    U ← UnionPolygons(P)
    F ← FilterByArea(U, params.min_area)
    if params.detailed then
        return F or SimplifyPolygons(F, params.tolerance)
    else
        Convex hulls around polygons. Merge overlapping hulls.
        return Hulls with area ≥ params.min_area_hulls
    end if
end procedure

procedure BuildParallelograms(points, speeds)
    P ← empty list
    for each (x, t), speed in points, speeds do
        Create a parallelogram centered at (x, t), tilted according to speed
        Append the parallelogram to P
    end for
    return P
end procedure
```

2 Construction of Parallelograms

Let (x, t) be the center of the parallelogram, v the speed in mph, L the total spatial length in feet, and T the total time span in seconds.

Convert speed to feet per second: $s = v \cdot 1.46667$

$$\Delta x = \frac{L}{2}, \qquad \Delta t = \frac{T}{2}$$

$$\begin{aligned} C_1 &= \big(t - \Delta t,\ x - s\Delta t - \Delta x\big) \\ C_2 &= \big(t - \Delta t,\ x - s\Delta t + \Delta x\big) \\ C_3 &= \big(t + \Delta t,\ x + s\Delta t + \Delta x\big) \\ C_4 &= \big(t + \Delta t,\ x + s\Delta t - \Delta x\big) \end{aligned}$$

The parallelogram is defined by the four corners C_1, C_2, C_3, C_4.