Field Mapping Task Splitting Algorithm FMTSA

Nishan Mainali

Intro

1. Factors to consider while doing field mapping
2. Algorithm steps
3. Queries ~~& visualization~~ (hand made drawings)

Factors to consider while doing field mapping

1. Geographic
2. Economic (cheaper & Faster)
3. Political
4. Cultural
5. Religious

Factors to consider while doing field mapping

1. **Geographic**
2. **Economic**
3. Political
4. Cultural
5. Religious

What I have chosen?

Geographic Factors

1. **Water bodies**: lakes, ponds, reservoirs, streams, or coastlines
2. **Soil types**: stability, erosion, or the presence of underground water
3. **Landforms**: mountains, hills, valleys, plains, plateaus, or coastal areas
4. **Vegetation**: forests, grasslands, wetlands, or agricultural areas
5. **Natural hazards**: earthquakes, landslides, floods, or wildfires
6. **Sensible areas**: military area and area of conflict
7. **Accessibility**: roads, trails, bridges, or airports

Economic Factors (Cheaper and Faster)

1. **Building density**: areas with higher cluster size
2. **Travel time**: aggregated distance between all the nodes

Algorithm steps includes:

1. Select area & retrieve OSM data
2. Analyze the data
3. Analyze geographic factors
4. Analyze economic factors
5. Define scoring (weight) criteria
6. Generate the splitted map
7. Validate and adjust
8. Assign tasks and execute field mapping

Step 1: Select area & retrieve OSM data

1. Area selection for field mapping

   Administrative boundaries

   OR

   Scope of your field mapping

2. Importing data using OSM2PGSQL

Step 2: Analyze and selection of data

1. POIs to collect
2. Listing related amenity tags
3. Remove irrelevant data (filter)

Step 3: Analyze geographic factors (setting priority & avoiding)

1. **Water bodies:** dissecting area if it affects accessibility
2. **Soil types:** removing area if it affects viability
3. **Landforms**: set low priority to the area if accessibility is difficult
4. **Vegetation**: set low priority to the area if accessibility is difficult & dangerous
5. **Natural hazards**: avoid hazard areas
6. **Sensitive areas**: avoid areas of conflict and non permissive
7. **Accessibility**: set low priority for areas with low accessibility

Query: Split the map into subsets based on **rivers**

```
INSERT INTO divided_subsets (geometry)
SELECT (ST_Dump(ST_Polygonize(riverlines))).geom
FROM (
  SELECT ST_Union(way) AS riverlines
  FROM planet_osm_line
  WHERE waterway IS NOT NULL
) AS rivers;
```

Query: Removing nodes for area where **soil types** has erosion, or the presence of underground water

```
DELETE FROM nodes
WHERE id IN (
    SELECT n.id
    FROM nodes n
    JOIN ways_nodes wn ON n.id = wn.node_id
    JOIN ways w ON wn.way_id = w.id
    JOIN way_tags wt ON w.id = wt.way_id
    WHERE wt.key = 'soil_type' AND wt.value IN ('erosion', 'underground_water')
);
```

Query: Removing nodes for area where **landform** is difficult to access

```
DELETE FROM your_nodes_table
WHERE id IN (
    SELECT n.id
    FROM your_nodes_table n
    JOIN your_tags_table t ON n.id = t.node_id
    WHERE t.landform IN ('beach', 'cliff', 'dune_system', 'esker')
);
```

Query: Removing nodes for area where **vegetation** is difficult to access

```
DELETE FROM nodes
WHERE id IN (
    SELECT n.id
    FROM nodes n
    INNER JOIN (
        SELECT DISTINCT node_id
        FROM (
            SELECT node_id, tags
            FROM planet_osm_point
            UNION ALL
            SELECT node_id, tags
            FROM planet_osm_polygon
        ) sub
        WHERE tags->'landuse' = 'forest'
        OR tags->'wetland' = 'swamp'
        -- Add more relevant tags here
    ) v ON n.id = v.node_id
);
```

Query: Removing nodes where **natural hazard** is present

```
DELETE FROM nodes
WHERE EXISTS (
    SELECT 1
    FROM planet_osm_point AS p
    WHERE ST_DWithin(nodes.geom, p.way, <distance_threshold>)
    AND (        p.natural = 'earthquake' OR
        p.natural = 'landslide' OR
        p.natural = 'flood' OR
        p.natural = 'wildfire'
    )
);
```

Query: Removing nodes for **sensitive area**

```
DELETE FROM your_map_table
WHERE EXISTS (
    SELECT 1
    FROM your_spatial_table sensitive_areas
    WHERE ST_Within(your_map_table.geometry_column, sensitive_areas.geometry_column)
    AND sensitive_areas.feature_type IN ('nuclear_power_plant', 'embassy', 'military_restricted', 'military_danger_area', 'military_barracks')
);
```

Step 4: Analyze economic factors (setting priority)

1. **Building density:** Higher building density may indicate areas with more amenities and potential POIs to collect.
2. **Travel time**: Areas with shorter travel times may be prioritized for efficient field mapping.

Query: Split map into subsets with higher **building density**

```
WITH node_counts AS (
  SELECT
    area_id,
    COUNT(*) AS total_nodes
  FROM
    your_osm_nodes_table
  GROUP BY
    area_id
),
area_subsets AS (
  SELECT
    a.area_id,
    a.geometry
  FROM
    your_osm_areas_table AS a
  INNER JOIN
    node_counts AS nc ON a.area_id = nc.area_id
  WHERE
    nc.total_nodes > <your_threshold_value>
)
SELECT
  asub.area_id,
  asub.geometry
INTO
  your_output_table
FROM
  area_subsets AS asub;
```

Query: Split map into subsets based on **connectivity**

```
CREATE TEMP TABLE connected_areas AS (
    SELECT a.id, a.geom
    FROM your_table_name a
    WHERE ST_Intersects(a.geom, (
        SELECT ST_Union(b.geom)
        FROM your_table_name b
        WHERE ST_Touches(a.geom, b.geom)
    ))
);
CREATE INDEX connected_areas_geom_idx ON connected_areas USING gist(geom);
CREATE TABLE connected_areas_permanent AS (
    SELECT *
    FROM connected_areas
);
ALTER TABLE connected_areas_permanent ADD PRIMARY KEY (id);
```

Step 5: Define scoring criteria

1. Defining scoring criteria to geographic and economic factors.
2. Scoring (weight) is based on importance of factors to the field mapping goals.
3. By default, all the scores (weight) are set to high for the range of particular factor.

Step 5: Define scoring criteria (Geographic factors)

Geographic Factors	**Weight/Score**	**Amenities and Tags**
Water bodies	0-10	* Lakes: natural=water + water=lake **(n)** * Ponds: natural=water + water=pond **(n)** * Reservoirs: water=reservoir **(n)** * Streams: waterway=stream **(n)** * Coastlines: natural=coastline **(n)**
Soil types	0-5	* Erosion: erosion=yes * Underground water: water=underground
Landforms	0-8	* Mountains: natural=peak * Hills: natural=hill * Valleys: natural=valley * Plains: natural=plain * Plateaus: natural=plateau * Coastal areas: natural=coastline

Step 5: Define scoring criteria (Geographic factors)

Geographic Factors	Weight/Score	Amenities and Tags
Vegetation	0-7	* Forests: landuse=forest * Grasslands: landuse=grass * Wetlands: natural=wetland * Agricultural areas: landuse=farmland
Natural hazards	0-9	* Earthquakes: natural=earthquake * Landslides: natural=landslide * Floods: natural=flood * Wildfires: natural=wildfire
Wildlife/biodiversity	0-6	* Wildlife: tourism=wildlife * Protected areas: boundary=protected_area * Important ecological sites: boundary=protected_area
Accessibility	0-10	* Roads: highway=primary/secondary/tertiary * Trails: highway=path * Bridges: bridge=yes * Airports: aeroway=aerodrome

Step 5: Define scoring criteria (Economic factors)

Economic Factors	**Weight/Score**	**Amenities and Tags**
Building density	0-10	* Commercial buildings: building=commercial * Residential buildings: building=residential * Public buildings: building=government
Travel time	0-8	* Highways: highway=primary/secondary/tertiary * Public transportation: public_transport=station

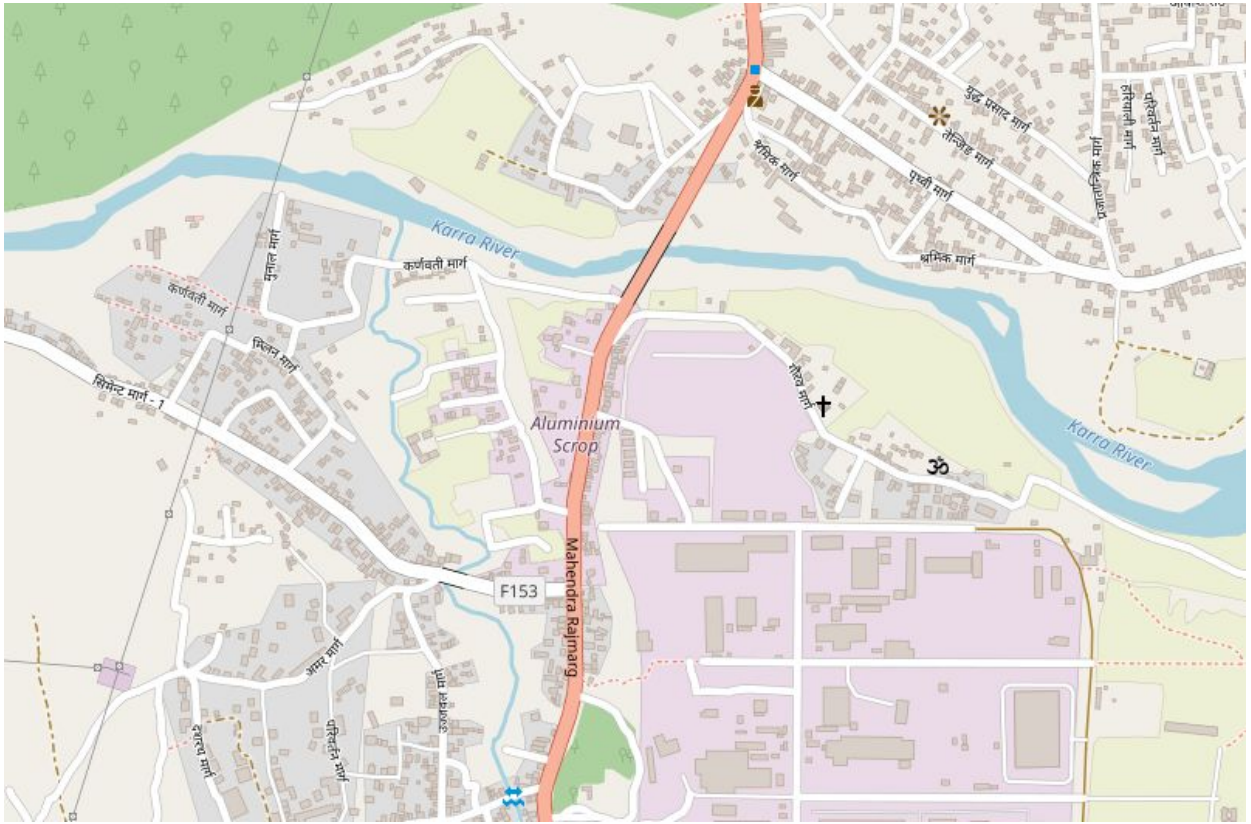
Step 6: Generate the splitted map

1. Defines minimum and maximum number of building for a splitted task (group of buildings) to be generated by algorithm.
2. Select and dissect the map using all the factors as mentioned in step 5 using the query.
3. Count the number of buildings in each area splitted area (created after dissecting)
    a. Use the centroid of building to count
    b. If centroid is used, buildings that are intersected by boundary can also be counted in an area and will not be missed.
4. If count of buildings is higher than “max. number of building” setted in point 1 then split the area into half until this(4) criteria gets fulfilled.
5. If count of buildings is lower than “min. number of building” setted in point 1 then merge the area with the nearest area until this(5) criteria gets fulfilled.
    a. If the area has to be merged then merge the area starting with lowest weight/score (mentioned in step 5) nearby because it will less affect the field mapping task.
6. Re-run from point 3 again until no action is taken either to split or merge.
    a. This will help re-evaluate the count of buildings changed during merge and split

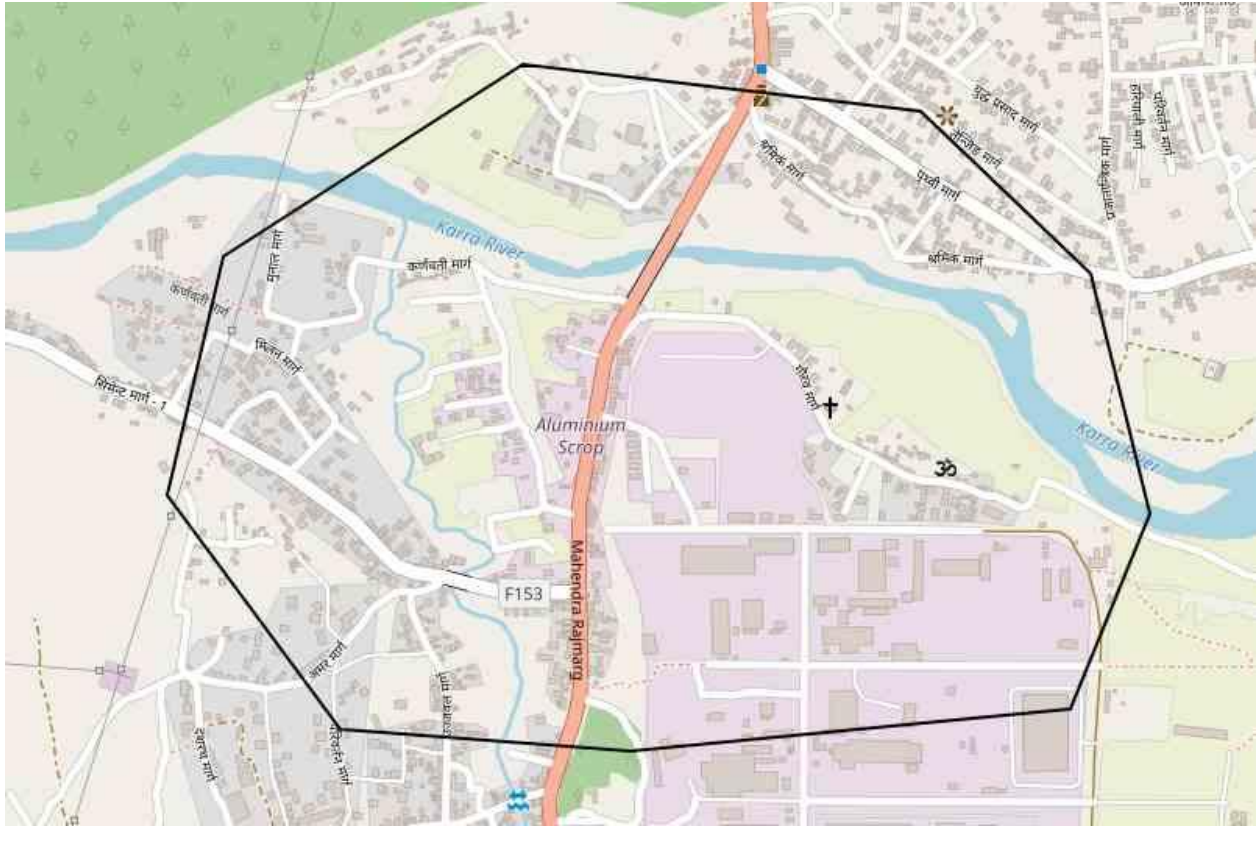
Step 7: Validate and adjust

1. Merge unnecessary split (human factor)
2. Draw custom areas for task (exceptional conditions)
3. Remove areas (exceptional conditions)

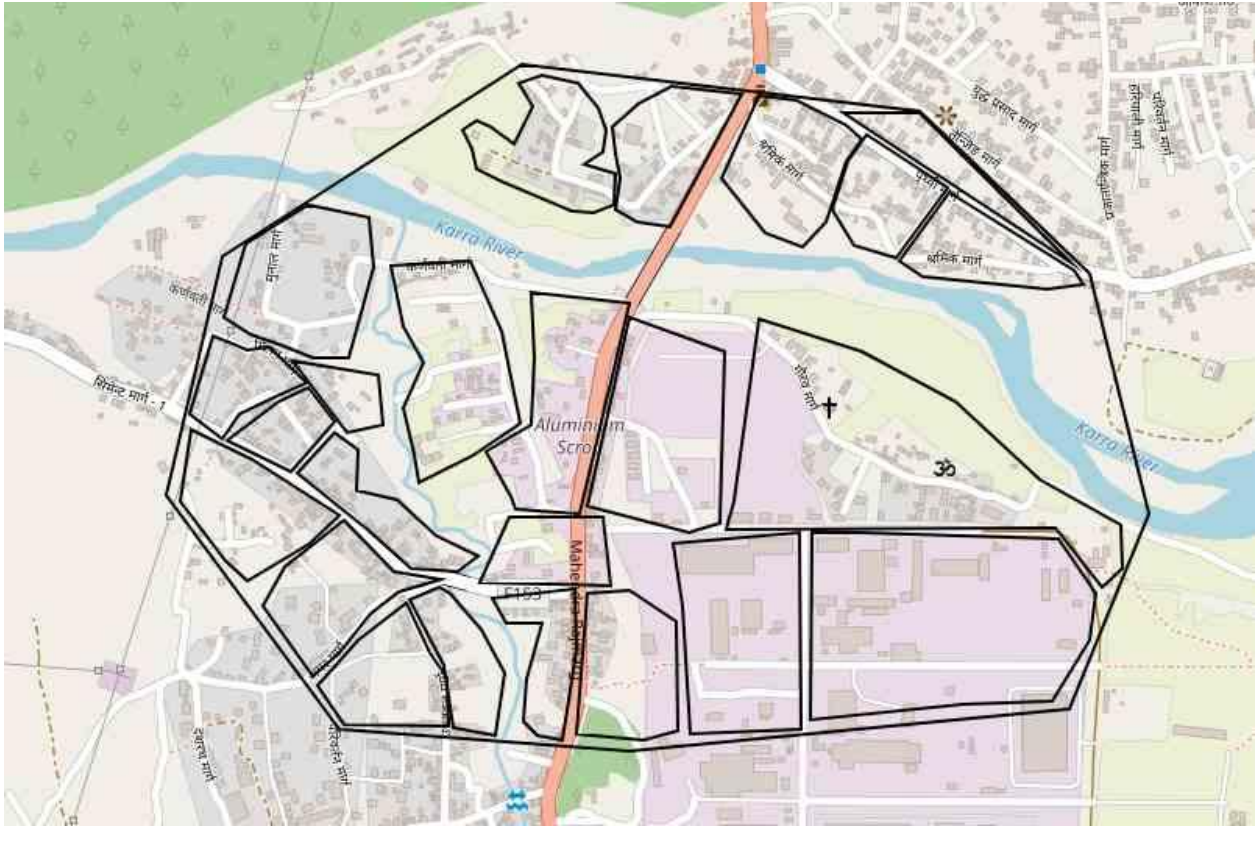
Demo: Region for Field Mapping



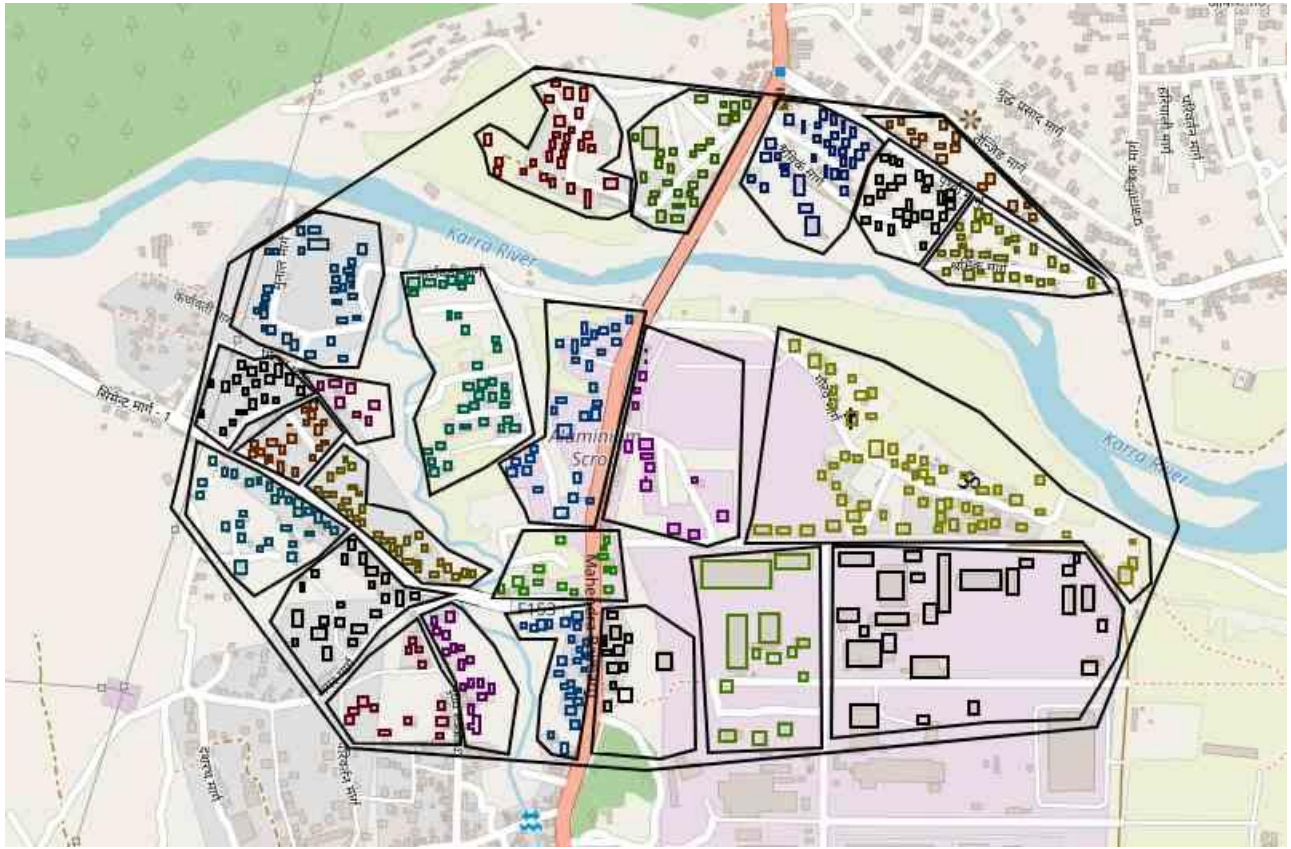
Selected Area for Field Mapping



Algorithm in action (Hand drawn - Not computer generated)



After running the algorithm



Final result

Step 8: Assign tasks and execute field mapping

1. Based on teams expertise and individual skills
2. With necessary technologies and equipment

Thank you for this opportunity!