Technical Report

Development of Automated Roadside Video Surveys for Detecting and Monitoring Coconut Rhinoceros Beetle Damage

Aubrey Moore

September 12, 2020

Contents

Listings

1 Recording Roadside Videos

2 Detecting CRB Damage in Video Frames

3 Creating a CRB Damage Survey Database

Have a look at Listing 1.

4 Visualizing Survey Results on a Map

Listing 1: create_tables.sql

```
--
    CREATE TABLE videos (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        name TEXT NOT NULL UNIQUE,
        device TEXT,
        video_app TEXT,
        camera_options TEXT,
        location_app TEXT,
        notes TEXT,
        gb FLOAT,
        fps FLOAT,
        resolution TEXT,
        lens TEXT,
        camera_mount TEXT,
        vehicle TEXT,
        camera_mount_position TEXT,
        camera_orientation TEXT,
        horizontal_angle FLOAT,
        vertical_angle FLOAT
    );

--
CREATE TABLE tracks (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    name TEXT NOT NULL UNIQUE,
    FOREIGN KEY(name) REFERENCES frames(name)
);

SELECT AddGeometryColumn('tracks', 'geometry', 3857, 'LINESTRING', 'XY');

--
CREATE TABLE frames (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    video_id INTEGER,
    frame_number INTEGER NOT Null,
    time TEXT,
    FOREIGN KEY(video_id) REFERENCES videos(id),
    UNIQUE(video_id, frame_number)
);

SELECT AddGeometryColumn('frames', 'geometry', 3857, 'POINT', 'XY');

--
CREATE TABLE trees (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    frame_id INTEGER,
    damage_index INTEGER NOT NULL,
    FOREIGN KEY(frame_id) REFERENCES frames(id)
);

SELECT AddGeometryColumn('trees', 'geometry', -1, 'MULTIPOINT', 'XY');

--
CREATE TABLE vcuts (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    frame_id INTEGER,
    FOREIGN KEY(frame_id) REFERENCES frames(id)
);

SELECT AddGeometryColumn('vcuts', 'geometry', -1, 'POLYGON', 'XY');
```

Listing 2: create_views.sql

```
-- Creates a view for use with QGIS
-- The geometry column contains camera location coordinates.
-- Note: SQL for this spatially enabled view was developed using spatiallite_gui Query/View Composer

CREATE VIEW "trees_view" AS
SELECT "a"."damage_index" AS "damage_index", "b"."ROWID" AS "ROWID", "b"."geometry" AS "geometry"
FROM "trees" AS "a"
JOIN "frames" AS "b" ON ("a"."frame_id" = "b"."id");

INSERT INTO views_geometry_columns
(view_name, view_geometry, view_rowid, f_table_name, f_geometry_column, read_only)
VALUES ('trees_view', 'geometry', 'rowid', 'frames', 'geometry', 1);

-- Creates a view for use with QGIS
-- The geometry column contains camera location coordinates.
-- Note: SQL for this spatially enabled view was developed using spatiallite_gui Query/View Composer

CREATE VIEW "vcuts_view" AS
SELECT  "b"."ROWID" AS "ROWID", "b"."geometry" AS "geometry"
FROM "vcuts" AS "a"
JOIN "frames" AS "b" ON ("a"."frame_id" = "b"."id");

INSERT INTO views_geometry_columns(
view_name, view_geometry, view_rowid, f_table_name, f_geometry_column, read_only)
VALUES ('vcuts_view', 'geometry', 'rowid', 'frames', 'geometry', 1);
```

Listing 3: create_grid.sql

```
BEGIN;

CREATE TABLE grid (id INTEGER PRIMARY KEY AUTOINCREMENT);

SELECT AddGeometryColumn('grid', 'geometry', 3857, 'MULTIPOLYGON', 'XY');

INSERT INTO grid (geometry)
        SELECT SquareGrid(Extent(geometry), 1000) FROM frames;


CREATE TABLE grid1 (id INTEGER PRIMARY KEY AUTOINCREMENT);

SELECT AddGeometryColumn('grid1', 'geometry', 3857, 'POLYGON', 'XY');

INSERT INTO grid1 (geometry)
        SELECT geometry
        FROM ElementaryGeometries
        WHERE f_table_name = 'grid'
                AND origin_rowid=1;


CREATE TABLE mean_damage_index (id INTEGER PRIMARY KEY AUTOINCREMENT, mean_damage_index DOUBLE);

SELECT AddGeometryColumn('mean_damage_index', 'geometry', 3857, 'POLYGON', 'XY');

INSERT INTO mean_damage_index (mean_damage_index, geometry)
        SELECT AVG(damage_index), grid1.geometry
        FROM trees_view, grid1
        WHERE Contains(grid1.geometry, trees_view.geometry)
        GROUP BY grid1.id;

--Clean up

DROP TABLE grid;

DROP TABLE grid1;

COMMIT;
```

Listing 4: create_crb_damage_map.py

```
def load_guam_osm():
    canvas = iface.mapCanvas()
    url = 'type=xyz&url=https://a.tile.openstreetmap.org/{z}/{x}/{y}.png&crs=EPSG3857'
    rlayer = QgsRasterLayer(url, 'Guam', 'wms')
    QgsProject.instance().addMapLayer(rlayer)
    rect = QgsRectangle(16098000.0, 1486000.0, 16137000.0, 1535000.0)
    canvas.setExtent(rect)
    canvas.update()


def load_layer_from_db(table_name):
    uri = QgsDataSourceUri()
    uri.setDatabase('/home/aubrey/Documents/populate_spatialite/videosurvey.db')
    schema = ''
    table = table_name
    geom_column = 'geometry'
    uri.setDataSource(schema, table, geom_column)
    display_name = table_name
    vlayer = QgsVectorLayer(uri.uri(), display_name, 'spatialite')
    QgsProject.instance().addMapLayer(vlayer)


def style_mean_damage_index():
    join_layer = QgsProject.instance().mapLayersByName(
        'mean_damage_index')[0]
    target_field = 'mean_damage_index'
    legend = [
        {'low': 0.0, 'high': 0.0, 'color': '#008000', 'label': 'No damage'},
        {'low': 0.0, 'high': 0.5, 'color': '#00ff00', 'label': '0.0 - 0.5'},
        {'low': 0.5, 'high': 1.5, 'color': '#ffff00', 'label': '0.5 - 1.5'},
        {'low': 1.5, 'high': 2.5, 'color': '#ffa500', 'label': '1.5 - 2.5'},
        {'low': 2.5, 'high': 3.5, 'color': '#ff6400', 'label': '2.5 - 3.5'},
        {'low': 3.5, 'high': 4.0, 'color': '#ff0000', 'label': '3.5 - 4.0'},
    ]
    myRangeList = []
    for i in legend:
        symbol = QgsSymbol.defaultSymbol(join_layer.geometryType())
        symbol.setColor(QColor(i['color']))
        myRangeList.append(QgsRendererRange(
            i['low'], i['high'], symbol, i['label'], True))
    myRenderer = QgsGraduatedSymbolRenderer(target_field, myRangeList)
    myRenderer.setMode(QgsGraduatedSymbolRenderer.Custom)
    join_layer.setRenderer(myRenderer)


# MAIN

load_guam_osm()
load_layer_from_db('tracks')
load_layer_from_db('frames')
load_layer_from_db('trees_view')
load_layer_from_db('vcuts_view')
load_layer_from_db('mean_damage_index')
style_mean_damage_index()
```