

Бібліотека для математичного аналізу оброблених земельних ділянок за даними з gps-трекера в json-форматі (Version №3)

Опис основних режимів роботи:

Дана бібліотека реалізована на мові C. Основні функції бібліотеки були сформовані так, щоб забезпечити роботу в двох режимах: **автоматичного** та **ручного**.

Введення цих двох режимів зумовлено тим, що бібліотека для обчислень використовує два внутрішні формати **.fld(field format)** та **.tmsg(tracker messages format)**. Оскільки дані передаються в json форматі, то в автоматичному режимі функції забезпечують автоматичне конвертацію файлів у форматі json у тимчасові файли у внутрішньому форматі представлення використання даних файлів для обчислень та автоматичне видалення тимчасових файлів. Функції ручного режиму - це функції за допомогою яких відбувається послідовне перетворення із одного формату в інший та проведення обчислень.

Основні функції автоматичного режиму

Для обчислень в автоматичному режимі використовуються такі функції:

json_field_size - обчислює площу геозони в гектарах використовуючи алгоритм апроксимації сіткою. Відповідно при обчисленнях необхідно вказувати розмір сітки в метрах для забезпечення необхідної точності (рекомендований розмір сітки 1метр).

Прототип функції:

double json_field_size(char *json_sourse, int field_id, double grid_size);

На вхід функція приймає шлях до файлу з геозонами в json форматі, id геозони та розмір сітки в метрах та повертає значення площі в гектарах.

json_polygon_size - ще один з способів обчислення площі геозони, в якому геозона представляється як N-кутний багатокутник.

Прототип функції :

double json_polygon_size(char *json_field_sourse,int field_id);

Функція приймає на вхід файл в json форматі з геозонами і id геозони.Функція повертає значення в гектарах.

json_field_perimeter — функція визначає периметр геозони і повертає значення в кілометрах.

Прототип функції:

double json_field_perimeter(char *json_field_sourse, int field_id);

Функція приймає на вхід файл з геозонами в json форматі та id геозони.

field_size_and_perimeter — функція обраховує периметр геозони та площу за допомогою апроксимації сіткою та повертає значення площі і периметру вибраної геозони.

Прототип функції:

void field_size_and_perimeter(char *json_field_file,int field_id ,double grid_size,double *area_size, double *perimeter);

Функція приймає на вхід файл з геозонами в json форматі, id геозони, розмір сітки, адреси змінних в які будуть занесені дані про площу геозони(га) та периметр(км).

json_field_analyse - аналізує дані з геозони і треку, які подаються в json форматі та обчислює сумарні площі оброблених ділянок геозони, необроблених та повторно оброблених ділянок.

Прототип функції:

void json_field_analyse(char *json_field_file,int field_id,char *json_message_file, double area_size, double width, double grid_size,double *used_field_size,double *not_used_field_size, double *poly_used_field_size);

На вхід функція приймає :

файл з геозонами в json-форматі;

id геозони;
файл з повідомленнями трекера в json форматі;
площу геозони в гектарах;
ширину прицепного ;
розмір сітки (оптимальний розмір половина ширини прицепного);
адреси змінних в які будуть занесені дані про площу обробленої ділянки , площу
необробленої ділянки та площу повторно обробленої ділянки.

Функції ручного режиму

В ручному режимі конвертація із json формату у внутрішні формати здійснюється окремими функціями ,також доступні функції для прямого обчислення та аналізу даних безпосередньо із файлів внутрішнього формату.

json_fields_to_fld - дана функція здійснює конвертацію з файла в .json форматі , у файл у форматі .fld .

Прототип функції на мові C.

void json_fields_to_fld(char *source_filename, int field_id, char *dst_filename);

Таким чином дана функція отримує на вхід шлях до файлу в json форматі, внутрішній id геозони в системі та шлях і назву файлу у форматі fld ;

json_messages_to_tmsg — дана функція здійснює конвертацію файла у json форматі , у файл в форматі .tmsg.

Прототип функції на мові C.

void json_messages_to_tmsg(char *source_filename_json, char *destination_filename_tmsg);

Функція приймає на вхід ім'я файлу у форматі json та ім'я файлу у форматі tmsg.

get_area - функція здійснює виокремлення тільки тих точок треку , які входять в задану геозону для подальшого аналізу треку.

Прототип функції:

void get_area(const char *track_path, const char *geofence_path, const char *area_path);

Дана функція приймає на вхід ім'я файлу з треком за певний період , у форматі tmsg, ім'я файлу з даними про геозону у форматі fld, та формує на основі даних файлів , файл в якому містяться лише ті точки треку, що знаходяться всередині геозони у форматі tmsg ;

reducing_points - функція , що здійснює розрідження точок геозони, в разі необхідності прискорення обчислень та зменшення точок геозони в ділянках де точки лежать на лініях з малим радіусом кривизни.

Прототип функції:

void reducing_points(const char *src_path, const char *dst_path, const double reducing_k);

Функція приймає ім'я файлу у форматі fld, та ім'я файлу , який буде відформатований у цьому ж форматі , та міститиме точки, які залишились в процесі розрідження. Такої функції необхідно передати аргумент від 0 до 1 , що вказує коефіцієнт розрідження. Чим менший коефіцієнт розрідження тим більше точок буде відсіяно в процесі розрідження.

field_size - функція , яка здійснює обчислення площі геозони в гектарах з певною точністю. Точність встановлюється довжиною сторони елемента сітки в метрах , яка накладається на геозону при обчисленнях. Рекомендована довжина сторони сітки 1 метр.

Прототип функції:

double field_size(char *field_path , double grid_size);

На вхід функція приймає ім'я файлу у форматі fld та розмір сітки в метрах для проведення обчислень. Функція вертає значення в гектарах.

Функція **area_analyser** - є одною з ключових , та здійснює обробку даних треку при цьому

обчислює оброблену площу поверхні , необроблену та повторну оброблену площу.

Прототип функції:

```
void area_analyser(char *field_path, char *area_path, double area_size, double width, double grid_size, double *used_size, double *not_used_size, double *poly_used_size);
```

Функція приймає на вхід шлях до файлу у форматі .tmsg в якому містяться лише точки треку, що лежать всередині геозони, також функція приймає шлях до файлу геозони у форматі fld, площу геозони в гектарах, ширину прицепного в метрах, розмір сітки(оптимальний розмір - половина ширини прицепного) в метрах, адреси змінних в які будуть занесені дані про : загальну оброблену площу(корисну площу) , необроблену площу , повторно оброблену площу.

Зауваження. Для коректного обчислення площі геозони , та аналізу за треком , точки геозони не повинні утворювати самоперетинів, інакше аналізатор може працювати некоректно.

field_perimeter - функція використовує файл геозони у форматі fld та обчислює периметр геозони в кілометрах.

Прототип функції:

```
double field_perimeter(char *fld_source);
```

Функція приймає на вхід ім'я файлу повним шляхом до нього у форматі fld та повертає значення периметру в кілометрах.

Функції що були додані в дану версію бібліотеки

Дана бібліотека повністю сумісна з версією 2.0 та наступною версією в якій було виправлено деякі помилки. Тому всі перераховані вище функції є присутні в даній версії бібліотеки .

Що нового в даній версії ?

В даній версії була додана можливість обрахування площ та периметру геозон у вигляді круга. Також був розроблений окремий аналізатор для даного типу геозон. Була додана окрема функція перевірки геозони на правильність. Правильною геозоною вважається така геозона , точки якої не утворюють самоперетинів та накладань . Після створення геозони здійснюється перевірка геозони на правильність .

Нові функції автоматичного режиму

До нових функцій бібліотеки , що ввійшли в дану версію належать:

- функція, що перевіряє геозону на правильність. Оскільки геозону можна умовно представити як багатокутник, то дана функція здійснює перевірку геозони на правильність за допомогою перевірки багатокутника на простоту. Простим вважається багатокутник, який не містить самоперетинів;

- функція, що здійснює обрахунок площі круглої геозони. Необхідним параметром є радіус геозони;

- функція для обрахунку периметра геозони.Необхідним параметром є радіус геозони;

- функція-аналізатор , що здійснює обрахунок обробленої площі, необробленої та повторно обробленої площі;

- функція , що здійснює перевірку геозони на правильність.

Нові функції ручного режиму

Всі нові функції ручного режиму аналогічні до нових функцій автоматичного режиму. Тому опис їх аналогічний. Варто зауважити, що функції ручного режиму передбачають обробку файлів даних у внутрішньому форматі (.tmsg, .fld).

Опис функцій автоматичного режиму

Функція обрахунку площі круглої геозони:

Прототип функції

double circle_geofence_square(double radius);

Функція приймає на вхід значення радіуса геозони в метрах та вертає значення площі геозони в гектарах.

Функція обрахунку периметру круглої геозони:

Прототип функції

double circle_geofence_perimeter(double radius);

Функція приймає на вхід значення радіуса геозони в метрах та вертає значення периметру геозони в метрах.

Функція аналізатор обробленої, необробленої та повторно обробленої площі ділянок.

Прототип функції

int json_circle_geofence_analyser(double center_longitude, double center_latitude, double radius, double width, double grid_size, char *json_path, double *processed, double *not_processed, double *poly_processed);

Функція обчислює значення площі і вертає код 1 в разі успішного завершення роботи та код ноль в разі помилки.

Вхідні параметри функції:

- довгота центра геозони,
- широта центра геозони,
- радіус геозони(в м),
- ширина прицепного(в м),
- розмір апроксимуючої сітки (в м),
- шлях до файлу треку в json - форматі,
- адреса змінної в яку буде занесений результат обчислення обробленої площі(га),
- адреса змінної в яку буде занесений результат обчислення необробленої площі(га),
- адреса змінної в яку буде занесений результат обчислення повторно обробленої площі(га).

Функція, що здійснює перевірку геозони на правильність.

Прототип функції

int json_is_simple_geofence(char *json_source_path, int id);

Функція приймає на вхід шлях до файлу з координатами геозон, та id геозони.

Функція вертає результат 1 якщо геозона є правильною і 0 якщо геозона містить самоперетини.

Опис функцій ручного режиму

Функція обрахунку площі круглої геозони:

Прототип функції

double circle_geofence_square(double radius);

Функція приймає на вхід значення радіуса геозони в метрах та вертає значення площі геозони в гектарах.

Функція обрахунку периметру круглої геозони:

Прототип функції

double circle_geofence_perimeter(double radius);

Функція приймає на вхід значення радіуса геозони в метрах та вертає значення периметру геозони в метрах.

Функція аналізатор обробленої , необробленої та повторно обробленої площі ділянок.

Прототип функції

int circle_geofence_analyser_tmsg(double center_longitude, double center_latitude, double radius, double width , double grid_size, char *track_path, double *processed, double *not_processed, double *poly_processed);

Функція обчислює значення площ і вертає код 1 в разі успішного завершення роботи та код ноль в разі помилки.

Вхідні параметри функції:

- довгота центра геозони,
- широта центра геозони,
- радіус геозони(в м),
- ширина прицепного(в м),
- розмір апроксимуючої сітки (в м),
- шлях до файлу треку в .tmsg - форматі,
- адреса змінної в яку буде занесений результат обчислення обробленої площі(га),
- адреса змінної в яку буде занесений результат обчислення необробленої площі(га),
- адреса змінної в яку буде занесений результат обчислення повторно обробленої площі(га).

Функція , що здійснює перевірку геозони на правильність.

Прототип функції

int is_geofence_simple(char *geofence_path);

Функція приймає на вхід шлях до файлу з координатами геозони в .fld-форматі.

Функція вертає результат 1 якщо геозона є правильною і 0 якщо геозона містить самоперетини.