

ТЗ на алгоритм автопереключения этажей

Основа алгоритма описана в ГуглДоке (https://docs.google.com/document/d/14-eeRuD7uKcdSLBYyXT-B3fKs9Ota3S_AwU4GdkTkQ/edit?usp=sharing proposed by Alexey Grechnev). По результатам исследований реальных данных к исходной версии был добавлен *feedback*-фильтр с коэффициентом **0,83**, а вектор весов установлен равным **[7 2 1]**.

Feedback-фильтр:

$$MLF_{filt}(t) = MLF_{filt}(t-1) * C_{filt} + MLF_{raw}(t) * (1-C_{filt})$$

где $MLF_{raw}(t)$ – текущее значение функции максимального правдоподобия для заданного этажа;

$MLF_{filt}(t)$ – текущее отфильтрованное значение функции максимального правдоподобия для заданного этажа;

C_{filt} – коэффициент фильтра (от 0 до 1); данный параметр нужно вынести в *config*-файл; значение по умолчанию равно 0,83.

Кроме того, для устранения ложных срабатываний была применена следующая идея - алгоритм должен переключать карту только в том случае, когда отфильтрованное значение функции максимального правдоподобия (ФМП) одного из этажей (отличного от текущего) последовательно N раз (по умолчанию **$N=3$** ; данный параметр необходимо вынести в *config*-файл) показывает наибольшее значение среди ФМП остальных этажей.

Описание алгоритма

currentMap -- текущая карта (на которой мы находимся);

calcMap -- карта вычисленная по нашей функции и RSSI пакетам на каждом шаге;

prevMap -- предыдущее значение *calcMap*;

counter – счетчик;

1. Рассчитываются ФМП для всех этажей, $f^i(BLE)$, от маяков которых приняты пакеты;

2. Если это первый момент времени, то определяется максимальное значение ФМП ($\max[f^i(BLE)]$) и по нему определяется текущий этаж – переменная *calcMap*; значение *currentMap* = *calcMap*; *prevMap* = *currentMap*, *counter* = 0;

3. Если не первый момент времени, то все рассчитанные значения ФМП, $f^i(BLE)$, поступают в *feedback*-фильтр.

4. По всем отфильтрованным значениям ФМП находится максимальное значение и, соответственно, этаж, который соответствует данной ФМП - *calcMap*;

5. Если prevMap != calcMap, то:
 counter = 0 (обнуляем, т.к. сменилась вычисленная карта с пред. шага)
 Если calcMap != currentMap, то ++counter;
 prevMap = calcMap;
6. Если prevMap == calcMap, то:
 Если currentMap == calcMap, то ++counter;
 Если counter == 3, то:
 переходим на карту calcMap;
 counter = 0;
 currentMap == calcMap;
 prevMap = calcMap;
7. Переходим к шагу 1.

Минус данного алгоритма: появляется задержка в переключении карты, примерно в 1 сек., если N=3.

Плюсы данного алгоритма: устраняются переключения карт из-за ложных срабатываний алгоритма (по крайней мере, на тех логах, которые есть в моем распоряжении на данный момент); решение становится приемлемым для практического использования.

Сравнительный анализ времени переключения карт для четырех тестовых случаев приведен в Табл. 1.

Таблица 1 – Сравнительный анализ времени переключения карт для четырех тестовых случаев на карте офиса Каа

Тестовые случаи	Случай	Реальный момент переключения (по видео)	Момент переключения / false det.
Тест 1а	«Коридор-офис»	20 сек.	23,67 / 0
Тест 1б	«Коридор-офис»	14 сек.	17,70 / 0
Тест 2а	«Офис-коридор»	15 сек.	19,46 / 0
Тест 2б	«Офис-коридор»	15 сек.	18,18 / 0