

УДК 004.514

А.В. Малютин, В.В. Любченко

## АЛГОРИТМЫ ГРУППИРОВКИ ГЕОСТАЦИОНАРНЫХ МЕТОК НА КАРТЕ GOOGLE MAPS

**Введение.** Результаты исследования компании Flutty показали, что средняя частота обращений пользователей к мобильным навигационным программам составляет 3.7 раза в неделю [1]. Несмотря на то, что это средний показатель, можно утверждать, что навигационные приложения играют важную роль в повседневном использовании мобильных устройств. Часто навигационные программы используются в весьма специфических условиях, например, когда пользователь управляет автомобилем, а навигационная программа прокладывает ему маршрут следования. В этом случае карта, отображаемая навигационной программой, должна быть с одной стороны читабельной, а с другой стороны – как можно более информативной. Для этого пользователю обычно показывают только ту информацию, в которой он сейчас нуждается, скрывая менее ценную в данный момент информацию. Группировка геостационарных меток на карте является одной из реализаций этого подхода. Геостационарная метка – это определенный объект, позиция которого задана координатами широты и долготы и этот объект отображается на карте с помощью определенного графического изображения.

В соответствии со стандартом ISO 25010, основными характеристиками качества программного обеспечения являются функциональность, надежность, удобство использования, быстродействие, возможность сопровождения и переносимость [2]. Для оценки эффективности алгоритма группировки, самой важной характеристикой является быстродействие, поскольку алгоритм должен оперативно отображать пользователю актуальное графическое представление меток на карте. Так же важна переносимость, поскольку программа, написанная по разработанному алгоритму будет запускаться на различных устройствах с различными диагоналями экранов при различных, динамически изменяющихся значениях масштаба карты.

**Цель работы.** Целью данной работы является разработка эффективного алгоритма графического отображения геостационарных меток, обеспечивающего быстродействие и переносимость для программного продукта в соответствии со стандартом ISO 25010.

**Основная часть работы.** Так как основная задача группировки меток – предотвращение графического наложения двух и более меток на карте, все используемые в настоящее время алгоритмы основаны на использовании порогового расстояния между двумя метками на экране. Если расстояние между двумя метками на экране меньше порогового значения, то над ними должна выполняться группировка; если расстояние между метками больше порогового значения – то разгруппировка.

Одним из первых алгоритмов группировки меток был алгоритм, основанный на квадратах [3]. Он реализуется путем разбиения карты на фиксированные области – квадраты. Каждый квадрат имеет по 4 значения координат, которые являются координатами углов квадрата на экране устройства. При этом диагональ каждого квадрата равна заданному пороговому значению. Сформулируем алгоритм:

1. Если существуют супер-метки, то выполнить их разгруппировку.
2. Разбить карту на квадраты, величина диагонали каждого квадрата определена заданным пороговым значением.
3. Определить, в каком квадрате лежит каждая пользовательская метка.
4. Если в одном квадрате лежат больше чем две метки, то выполнить их группировку в супер-метку.

Действия, описанные в данном алгоритме, выполняются каждый раз, когда пользователь меняет масштаб отображения карты. Оценка временной сложности алгоритма составляет  $O(n)$ , где  $n$  – количество пользовательских меток на карте.

Рассматриваемый алгоритм весьма прост в реализации, но имеет значительный недостаток. Рассмотрим пример (рис. 1). Расстояние между двумя метками меньше порогового значения, то есть нужно выполнить группировку двух меток. Однако они лежат в разных квадратах, поэтому при текущем масштабе группировка не будет выполнена. Такая ситуация очевидно не соответствует критерию переносимости и данный алгоритм не может считаться эффективным.

Рассмотрим еще один алгоритм – алгоритм группировки, основанный на расстоянии между метками на карте [3]. Для его реализации необходимо определить пороговое расстояние между двумя метками на карте.

Если существуют супер-метки, то перейти на шаг 2. Иначе, перейти на шаг 3.

Проанализировать все имеющиеся супер-метки. Если расстояние между двумя метками в супер-метке больше установленного порогового расстояния, то выполнить разгруппировку супер-метки.

Проанализировать все имеющиеся метки. Если расстояние между двумя метками меньше, чем

установленное пороговое расстояние, то выполнить группировку.



Рис. 1. Результат работы алгоритма группировки, основанного на квадратах

Действия, описанные в данном алгоритме, выполняются каждый раз, когда пользователь добавляет или удаляет метки с карты. Оценка временной сложности алгоритма составляет  $O(n^2)$ , где  $n$  – количество пользовательских меток на карте (независимо от того, принадлежит метка супер-метке или нет).

Данный алгоритм полезен, когда существует всего одно значение масштаба карты. В противном случае, группировка не будет выполняться при изменении масштаба отображения карты, что явно противоречит характеристикам переносимости.

Объединим полезные идеи двух рассмотренных алгоритмов. Мы видим, что группировка, основанная на вычислении расстояния между двумя метками, является более полезным подходом. Однако, с точки зрения реакции на масштабирование карты, расчет расстояния следует связывать с текущим масштабом, а не реальным расстоянием между метками. Таким образом, есть смысл ввести алгоритм группировки, который основан на относительном расстоянии между метками в системе координат экрана устройства. Мы получим неоспоримое преимущество: реализация имеет высокую переносимость на различных устройствах с различными разрешениями экранов. Кроме того, такой алгоритм нетрудно реализовать: необходимо лишь получить координаты точек на экране и затем сравнивать поочередно расстояние между ними с пороговым значением.

Сформулируем алгоритм группировки, основанный на расстоянии между метками на экране устройства.

Если существуют супер-метки, то перейти на шаг 2. Иначе, перейти на шаг 3.

Проанализировать все имеющиеся метки внутри каждой супер-метки. Если расстояние на экране между координатами супер-метки и метки, которая входит в состав данной супер-метки больше установленного порогового значения, то данная метка исключается из состава супер-метки. Расстояние между метками на экране вычисляется как евклидово расстояние:

$$DST = \sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2} \tag{1}$$

где  $x_1, y_1, x_2, y_2$  – координаты, соответственно, первой и второй точки на экране в пикселях. Выбор расстояния обусловлен свойствами пространства, в котором оно используется.

По окончании анализа отдельной супер-метки ведется проверка, осталось ли в ее составе еще метки. Если нет, то данная супер-метка удаляется.

Для каждой метки вычисляется расстояние между координатами этой метки и всех остальных меток. Если расстояние между двумя метками меньше порогового значения, то выполняется группировка этих двух меток в новую супер-метку.

Для каждой метки ведется сравнение расстояние между координатами этой метки и существующих супер-меток. Если расстояние между меткой и супер-меткой меньше порогового значения, то выполняется группировка выбранной метки в состав существующей супер-метки

Шаги, описанные в алгоритме, выполняются при смене пользователем масштаба отображения карты. При добавлении и удалении пользователем метки на карте, шаги алгоритма не выполняются и, следовательно, не происходит группировки и разгруппировки меток, так как пользователь должен убедиться в том, что он добавил или удалил метку корректно, следовательно, для этого остальные метки и супер-метки должны сохранить свое положение.

Для реализации данного алгоритма потребуются следующие структуры данных:

Класс, описывающий объект пользовательской метки, который содержит числовые переменные долготы и широты метки, а также строковую переменную, в которой будет содержаться описание данной метки.

Класс, описывающий объект пользовательской супер-метки, который содержит числовые переменные долготы и широты супер-метки, а также массив объектов пользовательских меток, которые входят в состав данной супер-метки.

Массив пользовательских меток карты, хранящий объекты пользовательских меток, которые пользователь добавил на карту.

Массив супер-меток карты, хранящий супер-метки, которые образовались в результате выполнения шагов алгоритма.

Так же, как и для алгоритма группировки, основанного на расстоянии между метками на карте, оценка временной сложности алгоритма составляет  $O(n^2)$ , где  $n$  – количество пользовательских меток на карте (независимо от того, принадлежит метка супер-метке или нет).

Были проведены исследования работы алгоритма на двух устройствах под управлением операционной системы Android с диагоналями экрана 4.7 и 9 дюймов. Положение метки на экране отображалось с помощью изображения, которое имело размеры 10x10 пикселей. Десять тестеров протестировали работу данного алгоритма поочередно на двух устройствах. В результате проведения эксперимента, 9 из 10 тестируемых отметили, что при расстоянии менее чем 80% от размера стороны картинки, метки становятся трудноразличимыми друг от друга. Следовательно, пороговое значение должно быть больше 8 пикселей и при этом не превышать размера стороны картинки. Поэтому, целесообразно установить пороговое значение в диапазоне 8-10 пикселей для эффективной работы алгоритма.

Рассмотрим результаты работы алгоритма группировки, основанного на расстоянии между метками на экране. На рис. 2 показан фрагмент карты с шестью пользовательскими метками красного цвета.

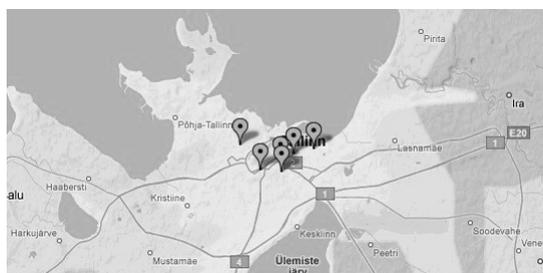


Рис. 2. Фрагмент карты с пользовательскими метками

После того, как пользователь изменяет масштаб отображения карты, выполняются действия, описанные в алгоритме и в результате этого на карте вместо трех меток появляется одна супер-метка черного цвета (рис. 3). Число 3 на графическом изображении супер-метки указывает на количество входящих в ее состав меток.



Рис. 3. Результат работы алгоритма группировки, основанного на расстоянии между метками на экране

**Вывод.** Таким образом, сформулированный алгоритм обеспечивает характеристики быстродействия и переносимости при работе с картой, оставаясь вычислительно рациональным. Несмотря на то, что алгоритм группировки меток разрабатывался с учетом того, что он будет реализован для Google Maps API в операционной системе Android, он легко может быть реализован и в других картографических системах.

#### ЛИТЕРАТУРА:

1. Запускалов А.В. Категории мобильных приложений и частота их использования [Электронный ресурс] / А.В. Запускалов. – Cossa.ru, 2012. – Режим доступа: <http://cossa.ru/articles/149/27855/>.
2. Wikipedia ISO/IEC 9126 [Электронный ресурс] / Wikipedia – en.wikipedia.org. – Режим доступа: [http://en.wikipedia.org/wiki/ISO/IEC\\_9126](http://en.wikipedia.org/wiki/ISO/IEC_9126)
3. Mahe L. Too Many Markers! [Электронный ресурс] / L. Mahe, C. Broadfoot. – Google Developers, 2010. – Режим доступа: <https://developers.google.com/maps/articles/toomanymarkers>.