

06, июнь 2016

УДК 004.021

Программный комплекс для обнаружения областей контроля на графе дорог

Савченко А.С., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»*

Научные руководители: Рудаков И.В., к.т.н, доцент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»*

Тассов К.Л., доцент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»*

bauman@bmstu.ru

Введение

С помощью систем фото-видеофиксации становится возможным не только фиксировать факт нарушения правил дорожного движения, но и раскрывать различные преступления, связанные с автомобилями, например, отслеживать транспортные средства подозреваемых. К поиску автомобилей, так или иначе, можно свести большой спектр задач, которые ежедневно возникают перед службами безопасности, начиная угонами, и заканчивая похищениями людей. На данный момент большая часть операций для поиска конкретного транспортного средства выполняется вручную. Поэтому, актуальной является автоматизация данной области. Для этого требовалось разработать алгоритм, который разобьёт дорожную сеть на области контроля, основываясь на информации о расположении комплексов фото-видеофиксации.

Назначение областей контроля

Область контроля – это простой цикл на графе дорог, который окружают комплексы фото-видеофиксации. Дорожная сеть представляет собой граф, в котором может быть очень большое число маршрутов. Оператор может просматривать данные с камер, которые заведомо не могли фиксировать искомый автомобиль. Это понижает скорость поиска – приходится просматривать лишние записи. Также, есть вероятность,

что оператор пропустит те камеры, которые могли бы зафиксировать транспортное средство. Это снижает точность поиска. Если дать оператору набор областей контроля, то он получит доступ сразу ко всем камерам, окружающим выбранную область, а не к отдельным камерам. Таким образом, в набор просматриваемых оператором записей попадают те, на которых автомобиль мог быть записан, и не попадают такие записи, на которых его заведомо не может быть.

На рис. 1 схематически изображена область контроля. Транспортное средство не может покинуть или въехать в нее, минуя комплекс фото-видеофиксации.

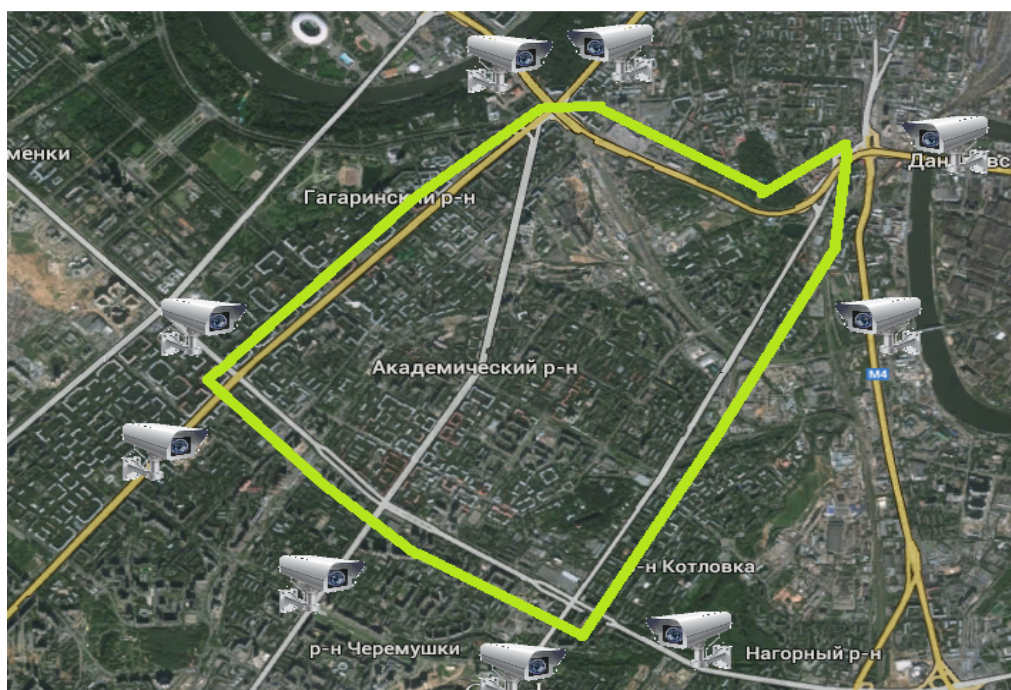


Рис. 1. Область контроля на карте

Проблема также состоит в том, что комплексы фото-видеофиксации размещали на самых оживленных улицах и трассах, так как первоначальной задачей стояло фиксирование нарушений ПДД, а не розыск какие-либо транспортных средств. Таким образом, даже на текущей, полностью функционирующей сети камер, не всегда удастся построить области контроля. Можно выделить вторую задачу – поиск потенциальных областей контроля, которые нуждаются в добавлении некоторого числа камер (потенциальных областей контроля). Следовательно, необходимо разработать алгоритм, который должен работать согласно IDEF0 диаграмме, приведенной на рисунке 2.

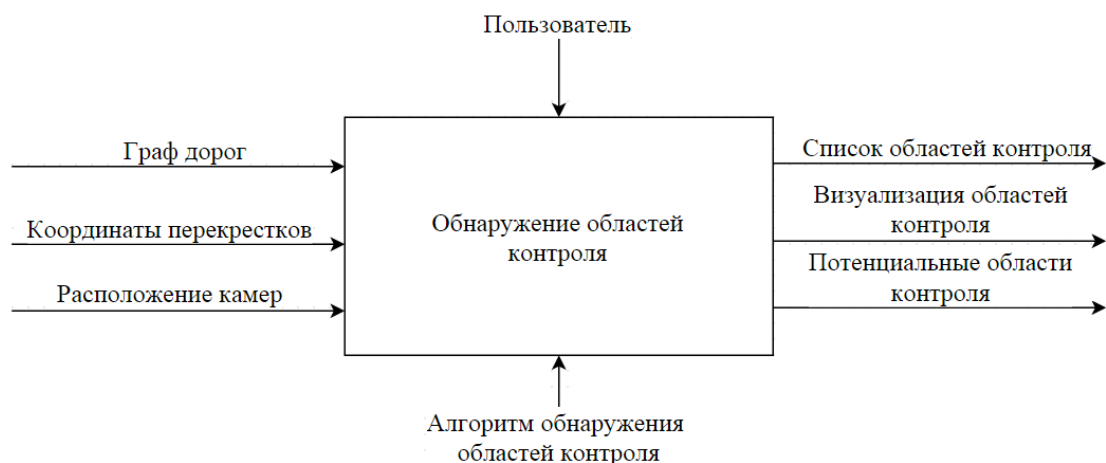


Рис. 2. Диаграмма IDEF0

Описание алгоритма поиска областей контроля

Область контроля – это простой цикл на графе дорог. Для решения задачи рассматривалась грубая модель дорожной сети – неориентированный граф. Следовательно, основным этапом разрабатываемого алгоритма является поиск простых циклов в неориентированном графе.

В ходе работы были изучены три алгоритма поиска простых циклов: алгоритм Тарьяна, алгоритм Шварцфитера и Лауэра, алгоритм Пейтона. Каждый из алгоритмов требует $O(V + E)$ памяти для хранения графа, где V – число вершин, а E – число ребер. В худшем случае сложность алгоритмов по времени для нахождения циклов в ориентированных графов равна [2]:

- Тарьян - $O(V * E * C)$.
- Шварцфитер и Лауэр - $O(V + E * C)$.
- Пейтон - $O(V^3)$.

Где V – число вершин, E – число ребер, а C – число простых циклов в графе.

Худший показатель достигается на графах с особой структурой, поэтому на реальных задачах алгоритм с более высокой сложностью может превзойти алгоритм с более низкой сложностью [3]. В результате тестирования алгоритмов для поиска простых циклов получился следующий список (от быстрого к медленному) [2]:

- Шварцфитер и Лауэр.
- Тарьян.
- Пейтон.

Несмотря на то, что алгоритм Шварцфитера и Лауэра обладает сравнимой скоростью с алгоритмом Тарьяна, он является самым трудным для реализации. Алгоритм

Пейтона способен обрабатывать графы с кратными ребрами – такая особенность не требуется в решаемой задаче. За основу был взят алгоритм Тарьяна, поскольку он прост в реализации и обладает хорошей скоростью выполнения.

Важно отметить, что рассмотренные алгоритмы применимы лишь к ориентированным графам. В решаемой задаче области контроля строились на неориентированных графах, количество циклов в которых может быть намного больше, чем в ориентированных [4]. Для того, чтобы алгоритм Тарьяна мог быть применен к неориентированным графам, были написаны следующие функции:

- Инвертирование – операция, в результате которой вершины цикла записываются в другом порядке.
- Нормализация – операция, в результате которой вершины цикла переставляются таким образом, чтобы вершина с наименьшим индексом стояла на первом месте.

Нормализация и инвертирование применяются к каждому найденному циклу и позволяют исключить хранение одинаковых циклов, вершины которых записаны в разном порядке. Если список найденных циклов не содержит нормализованной и инвертированной версии найденного цикла – то этот цикл объявляется новым и сохраняется. В результате работы алгоритма Тарьяна на графе будет найдено множество простых циклов. Далее необходимо определить, какие из этих циклов могут быть областями контроля.

Комплексы фото-видеофиксации на графе дорог смоделированы с помощью взвешенных ребер, значение 1 означает, что на дуге есть комплекс фото-видеофиксации, а 0 – его отсутствие. Необходимо найти такие циклы, которые были бы окружены ребрами с весом 1.

На этом же этапе обнаруживаются потенциальные области контроля. Пользователь может передать алгоритму число, равное допустимому недостатку комплексов фото-видеофиксации для каждой замкнутой области. Таким образом, если пользователь введет число 0, он получит все полноценные области контроля. Если же он введет 2 – он получит все замкнутые области, которые могли бы стать областями контроля при добавлении на их границы 2 комплексов фото-видеофиксации (т.е. потенциальные области контроля).

Чтобы проверить найденный простой цикл, для каждой вершины требуется:

1. Найти все вершины, которые доступны из данной за переход по одному ребру. Такие вершины считать соседями.

2. Рассматривать лишь те вершины, которые находятся вне цикла. Данная проверка производится по координатам – решается задача принадлежности точки к многоугольнику. Также исключаются вершины, которые являются вершинами данной замкнутой области.

3. Проверить ребра, соединяющие данную вершину и её соседей из пункта 2. Если на этом ребре нет камеры, то увеличить счетчик недостатка камер.

4. Если счетчик недостатка камер больше, чем допустимое значение, то замкнутая область не является областью контроля, иначе – является.

Для того, чтобы выделить все области контроля, необходимо применить алгоритм проверки к каждому найденному простому циклу.

Разработанный алгоритм состоит из модификации алгоритма Тарьяна и проверки найденных простых циклов. Найденную область контроля можно увидеть на рисунке 2. Все ребра, окружающие простой цикл, помечены единицей, кроме одного. Можно сделать вывод, что найдена потенциальная область контроля, и если на ребро {15, 1} будет установлена камера, то найденный цикл станет полноценной областью контроля.

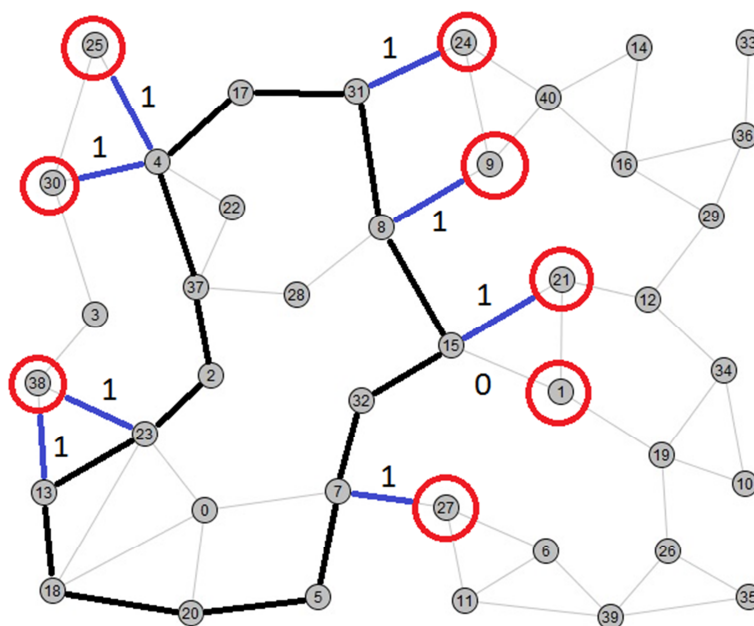


Рис. 4. Найденная потенциальная область контроля

Программный комплекс для поиска областей контроля

Был разработан программный комплекс, реализующий описанный алгоритм. Комплекс должен быть легко расширяемым и масштабируемым, исходя из этих соображений для разработки был выбран язык Java 7. Язык позволил производить обход графов с достаточной скоростью, и при этом предоставит широкий спектр возможностей

для работы с базой данных и для расширения программного комплекса. Для хранения данных была выбрана СУБД Oracle 10g Express Edition.

При разработке системы использовались следующие внешние библиотеки и платформы:

1. JUNG – хранение и обработка графов.
2. Swing – Графический интерфейс пользователя.
3. JUnit – тестирование.

Из рассмотренных библиотек необходимо выделить JUNG. В системе производится интенсивная обработка графов, поэтому необходимо во время исполнения хранить его в структуре данных, которая предоставляла бы широкий спектр функций для обработки графа и была бы достаточно быстрой. В качестве структуры для хранения графа была выбрана Java Universal Network/Graph Framework (JUNG).

Разработанная система состоит из следующих модулей и компонент (рис. 4):

1. Представление графа (На основе JUNG).
2. Обработчик графа.
3. Графический интерфейс пользователя.
4. Модуль хранения графа дорог.
5. Хранилище графов и найденных областей контроля.

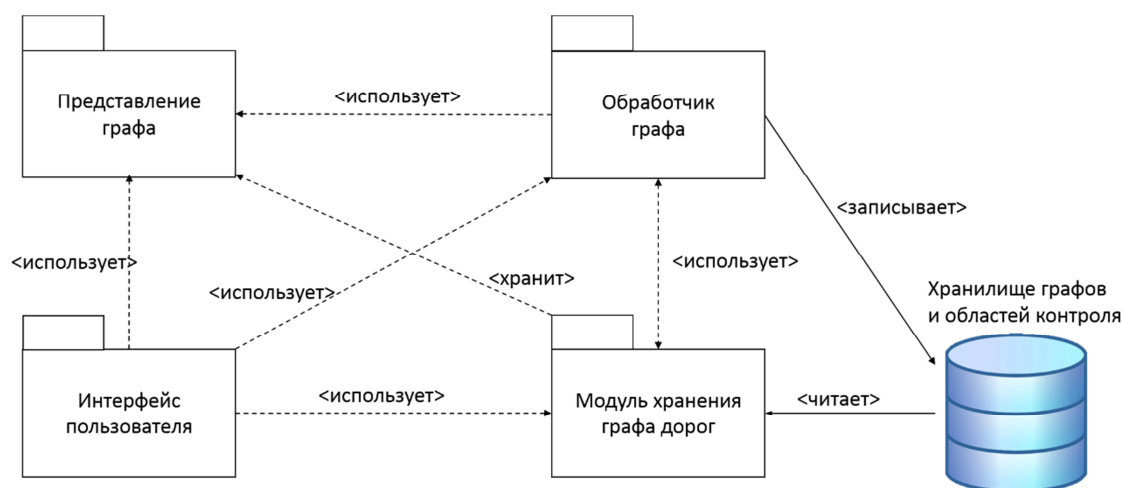


Рис. 5. Структура разработанного ПО

Сгенерированные и обработанные графы система позволяет сохранять в базу данных, и в любой момент загружаться из неё. Помимо самого графа и информации о расположении комплексов фото-видеофиксации, в базу также сохраняются все области контроля, которые были найдены в результате работы алгоритма, таким образом на больших графах не обязательно заново пересчитывать области контроля (расчет может

занимать около часа и более), а достаточно их сохранить в базу, и выгрузить при необходимости.

Разработанный программный комплекс требователен к системным ресурсам и быстродействию системы в целом. Это связано с вычислительной сложностью разработанного алгоритма Тарьяна, используемого для поиска простых циклов. Этот процесс является самым затратным этапом всего алгоритма выделения областей контроля. Время работы алгоритма сильно зависит от количества вершин и ребер в графе, поскольку он основан на поиске в глубину. Для ускорения работы алгоритма в программе предусмотрен критерий оптимизации – ограничение на длину каждого простого цикла. Необходимо сравнить и проанализировать полученные результаты с использованием критерия и без него.

Результаты исследования

Целью исследования является определение влияния критерия оптимизации на точность алгоритма и на скорость его выполнения.

Вершины / дуги	Ограничение на длину циклов	Время работы	Найденные области
68 / 103	-	4 мин 21 сек	6
68 / 103	13	0 мин 19 сек	6
68 / 103	12	0 мин 10 сек	5
90 / 132	-	8 мин 44 с	40
90 / 132	15	1 мин 22 с	28
90 / 132	10	0 мин 2 с	13

Первый из исследуемых графов состоял из 68 вершин и 103 ребер. Такая размерность является крайне высокой для поиска в глубину. В результате первого эксперимента было выявлено эталонное время обработки графа, которое в дальнейшем улучшалось вводом критерия оптимизации. В качестве такого критерия рассматривалось ограничение размера простых циклов. Алгоритм основан на поиске в глубину с рекурсивной реализацией. В таком случае, можно ограничить глубину рекурсии – это и будет ограничение размера цикла. То есть пользователь сможет задать число, которое будет больше или равно числу вершин в любом полученном простом цикле. Таким

образом сильно сокращается время выполнения алгоритма, а также отсеиваются заведомо лишние области контроля для пользователя. К примеру, если пользователь хочет найти лишь маленькие области размером в 5 вершин, ему не придется ждать, пока поиск в глубину будет погружаться в рекурсию тысячи и десятки тысяч раз. Это решение дает возможность оптимизировать разработанный алгоритм как с точки зрения скорости его выполнения, так и с точки зрения удобства для пользователя.

В таблице наглядно видна зависимость времени работы алгоритма от размера критерия оптимизации. Чем меньше ограничение на длину циклов – тем менее глубоко уходит в рекурсию алгоритм, таким образом получается значительный выигрыш в скорости. Но при этом не будут рассмотрены области больших размеров, и поэтому имеется погрешность в виде областей контроля, упущенных из рассмотрения в силу применения критерия оптимизации.

Заключение

Разработан алгоритм обнаружения областей контроля, которые должны увеличить скорость и точность поиска транспортных средств с помощью комплексов фото-видеофиксации. Алгоритм был исследован, был введен критерий оптимизации и исследовано его влияние на точность и скорость работы алгоритма. Также был разработан программный комплекс, состоящий из настольного приложения и базы данных, который позволил провести эксперименты.

Список литературы

- [1]. Белоусов А.И., Ткачев С.Б. Дискретная математика: учеб. для вузов /под ред. В.С. Зарубина, А.П. Крищенко. 3-е изд. М.: Изд-во МГТУ им. Н.Э. Баумана, 2004. 744 с. (Сер. Математика в техническом университете; Вып XIX).
- [2]. Niographs – A JAVA library of graph algorithms. Available at <https://code.google.com/p/niographs> (accessed 13.06.2014).
- [3]. Reactorweb. Finding all polygons in an undirected graph. Available at <http://blog.reactoweb.com/2012/04/algorithm-101-finding-all-polygons-in-an-undirected-graph>, accessed 16.06.2014.
- [4]. Alan Tucker. Applied Combinatorics, Hoboken, New Jersey: Wiley, 2012. 496 p.