

Выделение контуров зданий и распознавание служебных символов для трехмерной реконструкции объектов городской обстановки по топографическому плану

Игорь Соловьёв

Школа естественных наук, Дальневосточный федеральный университет, Владивосток, Россия

ИАПУ ДВО РАН, Владивосток, Россия

igorek.solovyev@mail.ru

Аннотация

В данной работе рассматривается метод реконструкции объектов городской обстановки по топографическому плану. Предлагается новый алгоритм распознавания служебных символов на топлане, содержащих информацию о типе здания и количестве этажей. Также, предлагается модификация волнового алгоритма, используемого для нахождения границ здания.

Ключевые слова: Распознавание, Реконструкция, Топоплан.

1. ВВЕДЕНИЕ

Создание трёхмерных моделей реальных объектов остаётся актуальной проблемой компьютерной графики и машинного зрения. Одной из важных практических задач является реконструкция объектов городской обстановки.

На сегодняшний день уже разработан ряд методов, направленных на решение поставленной задачи [3]. Одним из таких методов является реконструкция городского пространства по топографическому плану. При таком подходе необходимо определить геометрическую форму основания здания, а также распознать надпись содержащую информацию о типе здания и количестве этажей. Автоматическое выделение контура здания производится с помощью метода, описанного в [2]. Получение информации об этажности производится путём сравнения выделенной надписи с базой эталонов.

2. НАЧАЛЬНЫЕ ДАННЫЕ

Топоплан представляет собой чёрно-белое изображение, содержащее контуры зданий, улиц, информацию о количестве этажей зданий, данные о ландшафте, различные городские объекты и т.д.

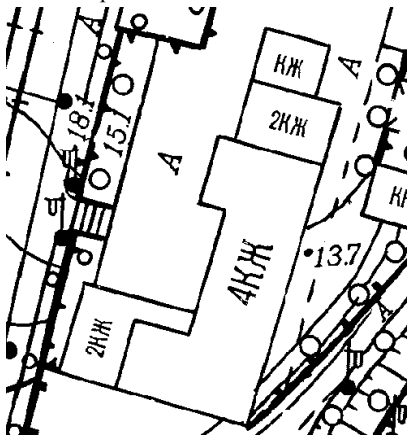


Рис. 1: Топографический план.

Преобразуем данное изображение в матрицу M , заполняя её нулями и единицами. Ноль соответствует белому цвету на топлане, а единица – чёрному.

3. ВОЛНОВОЙ АЛГОРИТМ

В рассматриваемом методе используется волновой алгоритм [1], однако скорость его выполнения не так высока, поэтому скорость работы распознавания также будет небольшой, особенно при анализе топлана больших размеров. Поэтому для интерактивной работы приложения существует необходимость в ускорении волнового алгоритма.

Использование данного алгоритма имеет одно преимущество. После его работы получаются данные о контуре и служебной информации. Причём эти данные едины, т.е. нет необходимости в том, чтобы устанавливать соответствия между символами и контуром, к которому они относятся.

3.1 ОПИСАНИЕ ВОЛНОВОГО АЛГОРИТМА

Волновой алгоритм довольно прост в реализации. Пусть на некоторой прямоугольной сетке все элементы равны либо 0 (свободные клетки), либо 1 («стены»). Выбирается любая свободная ячейка и ей присваивается значение $t_0 > 1$. Далее, сканируется вся сетка и если находится клетка со значением t_0 , то проверяются её соседи. Каждой пустой соседней ячейке присваивается значение $t_0 + 1$. Если какая-нибудь соседняя ячейка является «стеной», то ей присваивается значение -1 . Далее, вся сетка сканируется вновь, только на этот раз ищутся ячейки со значением $t_0 + 1$ и т.д. Алгоритм повторяется до тех пор, пока в замкнутой области, окруженной «стенами» не останется пустых ячеек.

Скорость работы такого алгоритма напрямую зависит от числа ячеек. Если количество элементов сетки равно N , то за t шагов волновому алгоритму придётся проверить $N \cdot t$ клеток. В худшем случае, при большом количестве пустых ячеек и одинаковых размерах сетки, число элементов будет равно $N = (2t + 1)^2$. Тогда алгоритму придётся обработать S_1 ячеек:

$$S_1 = (2t + 1)^2 t = 4t^3 + 4t^2 + t \quad (1)$$

Как правило, волновой алгоритм применяется для задач поиска кратчайшего пути в лабиринте. В нашем случае, путь не играет никакого значения, нам важно только воздействие алгоритма на рабочую область.

3.2 МОДИФИКАЦИЯ АЛГОРИТМА

Модификация алгоритма использует тот факт, что при поиске очередной ячейки нет необходимости в сканировании всей сетки, достаточно только проверить

соседние элементы уже рассмотренных ячеек. Таким образом, ускорить время работы алгоритма можно при помощи четырёх дополнительных переменных **L**, **T**, **R**, **B**, которые являются соответственно левой, верхней, правой и нижней границами некоторого окна поиска. Каждая переменная имеет значение минимального или максимального номера клетки по своему направлению (номера **Li** и **Ri** имеют значения минимального и максимального номеров по горизонтали, а номера **Ti** и **Bi** - по вертикали). Таким образом, требуется сканирование не всей сетки, а только лишь окна размером $(R-L) \cdot (B-T)$. Если на каком-то шаге волны обнаруживается пустая ячейка, то она проверяется на то, выходит ли она за рамки окна, если так, то соответственная граница сдвигается.

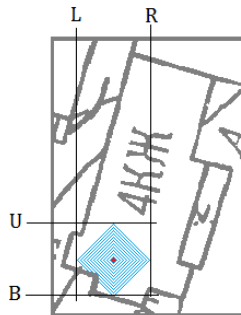


Рис2: Процесс распространения волны в окне.

Худшим вариантом для алгоритма будет ситуация, при которой начальной точку t_0 будет окружать множество пустых ячеек. В этом случае придётся сканировать окно размером 3×3 , на следующем шаге 5×5 и т.д. На шаге t окно сканирования будет иметь размеры $(2t+1) \times (2t+1)$. Таким образом максимальное число ячеек S_2 , которое необходимо просканировать будет определяться по формуле

$$S_2 = \sum_{k=1}^t (2k+1)^2 = \frac{4t^3 + 12t^2 + 11t}{3} \quad (2)$$

Ускорение, которое будет давать модифицированный алгоритм, определится как $a = S_1/S_2$.

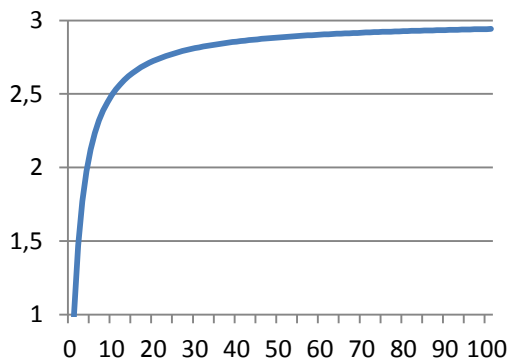


Рис. 3: График ускорения работы модифицированного алгоритма. Слева расположено ускорение, снизу – высота и ширина топоплана.

Таким образом, применение модификаций к волновому алгоритму позволяет ускорить алгоритм в 3 раза. Это особенно полезно при работе с сеткой больших размеров.

4. ВЫДЕЛЕНИЕ СЛУЖЕБНОЙ ИНФОРМАЦИИ

На топоплане информация о количестве этажей здания находится внутри его контура. Прежде, чем приступить к распознаванию этой информации, её необходимо выделить из всего набора тех данных, что находится на изображении.

Выберем любую нулевую точку матрицы **M**, причем так, чтобы она находилась внутри контура, образованного единичными элементами. Из этой точки запустим волновой алгоритм. После его работы матрица **M** изменится следующим образом:

- элементы, совпадающие с границей контура, будут равны -1;
- элементы, находящиеся на границе служебной информации, также будут равны -1;
- все точки, лежащие внутри контура и не принадлежащие служебной информации, будут равны числам больше 1.

Имеется два набора данных, отмеченных числом -1, поэтому необходимо их разделить. Применим к матрице **M** несколько фильтров, чтобы оставить только точки, принадлежащие служебной информации.

Первый фильтр удаляет точки границы контура. Для этого построчно перебираются все элементы матрицы **M**. Если найден элемент со значением -1, то он будет являться источником для расширенного волнового алгоритма по контуру. Действие такого алгоритма будет отличаться от рассмотренного выше следующими положениями:

- волна распространяется не по четырем направлениям, а по восьми;
- «пустыми» клетками будут клетки со значением -1;
- «стенами» будут все ячейки, значения которых больше нуля.

Результатом применения алгоритма будет отсутствие в матрице **M** элементов границы контура. Таким образом, все элементы матрицы равные -1 будут именно точками служебной информации.

Далее, перебираются все элементы матрицы. Если значение ячейки не равно -1, то из неё будет запущен волновой алгоритм, работающий по следующему принципу:

- волна распространяется по четырем направлениям;
- «пустыми» клетками будут те, значение которых не равно -1;
- «стенами» будут все ячейки, значения которых равно -1.

После применения такого алгоритма в матрице **M** образуются элементы со значением больше 1. Такие элементы никак не относятся к служебной информации. Проведённое действие позволило отделить нулевые элементы внутри служебной информации от тех, которые лежат вне её. Таким образом, элементы матрицы **M** равные 0 и -1 будут тем набором данных, который необходимо распознать как текст. Для простоты, изменим значения ячеек следующим образом:

- Если значение элемента равно 0 или -1, то ей присваивается значение 1;
- Если значение элемента больше 1, то ей присваивается значение 0.

Требуемый набор данных сформирован.

5. РАЗБИЕНИЕ НА СИМВОЛЫ

Перед распознаванием, полученный набор данных требуется разбить на отдельные символы. Для этого поочерёдно перебираются все элементы матрицы **M**. Если элемент равен 1, то из него запускается волновой алгоритм. «Пустыми» клетками для него будут клетки со значением 1, а «стенами» - клетки со значением 0. После этого, все элементы, значение которых больше 1, становятся частью другой матрицы **J₀**. Размеры такой матрицы соответствуют размеру области действия алгоритма. Элементы матрицы **M** со значением больше 1 меняют значение на 0. Повторяется процесс поиска единичного элемента. После его отыскания снова запускается волновой алгоритм и формируется матрица **J₁**. Таким образом создаются матрицы символов **J_k**, где **k** принимает значения от 0 до **w-1**.

6. ФОРМИРОВАНИЕ БАЗЫ ЭТАЛОНОВ

Каждым эталоном является один из символов служебной информации, который находится под одним из возможных угловповорота. Пусть топоплан содержит **n** служебных символов. В данном случае служебными символами будут именно те, которые находятся внутри контуров зданий. Ориентация топоплана может быть различной, следовательно ориентация символов тоже. В связи с этим количество вариантов каждого символа будет определяться формулой:

$$z = \text{round}\left(\frac{360^\circ}{\alpha}\right), \quad (3)$$

где α – минимальный угол поворота одного символа, выраженный в градусах. Каждый *i*-й вариант эталона будет повернут на угол $\alpha \cdot i$, где *i* принимает значения от 0 до **z-1**.

База эталонов будет представлять из себя матрицу **V** с размерами **n·z**. Каждый элемент такой матрицы содержит в себе данные об эталоне. **V_{ij}** также является матрицей, где *i* принимает значения от 0 до **n-1**, а *j* – от 0 до **z-1**. Такая матрица заполнена 0 и 1, 0 соответствует чёрному цвету эталонного изображения, а 1 – чёрному.

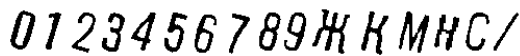


Рис. 4: Эталон служебных символов.

7. РАСПОЗНАВАНИЕ

Распознавание матрицы **J_k** будет производиться путём сравнения её с каждым эталоном из **V**. Пусть матрица **J_k** имеет размеры **WxH**, а размеры матрицы эталона **V_{ij}** равны **FxG**. Тогда между координатами элемента матрицы **J_k** и координатами матрицы эталона существует соотношение в виде:

$$\begin{aligned} x &= \frac{W}{F} \bar{x} \\ y &= \frac{H}{G} \bar{y} \end{aligned} \quad (4)$$

где *x* и *y* – координаты некоторого элемента матрицы **J_k**, а \bar{x} , \bar{y} – соответствующие им координаты матрицы эталона **V_{ij}**. Если элементы матриц с такими координатами равны, то процент их схожести увеличивается. Таким образом, перебираются все эталоны и выбирается тот, чей процент схожести выше. Процент схожести определяется путём поэлементного сравнения матриц:

$$P = \frac{d}{W \cdot H} * 100\%, \quad (5)$$

где **P** – процент схожести, **W**, **H** – соответственно ширина и высота символа **J_k**, **d** – количество элементов, для которых справедливо равенство:

$$J_{xy} = V_{\bar{x}\bar{y}} \quad (6)$$

При таком подходе, число сравнений равно произведению **n·z·w**, где **n** – число эталонов, **z** – число вариантов одного эталона, **w** – количество символов, которое требуется распознать.

8. РЕЗУЛЬТАТЫ

Представленный алгоритм был применён на топоплане рис. 5. Размер изображения 1550x1222 пикселей. Топоплан содержит 137 символов той служебной информации, которую необходимо распознать. Служебная информация состоит из 16 различных символов. Средние размеры всех эталонов 25x36 пикселей.

Результаты работы алгоритма представлены в таблице 1.

Таблица 1.

| α, градусы | Время загрузки, секунды | Время распознавания, секунды | Правильно распознано | % |
|------------|-------------------------|------------------------------|----------------------|------|
| 0,1 | 543 | 381,8 | 135 | 98,5 |
| 1 | 59,8 | 44,2 | 132 | 96,3 |
| 2 | 29,4 | 23,5 | 129 | 94,1 |
| 3 | 24,2 | 16,0 | 126 | 91,9 |
| 4 | 14,7 | 13,2 | 121 | 88,3 |
| 5 | 11,4 | 9,1 | 115 | 83,9 |
| 6 | 11,2 | 8,9 | 113 | 82,4 |
| 10 | 7,1 | 7,4 | 104 | 75,9 |
| 30 | 5,2 | 5,5 | 98 | 71,5 |

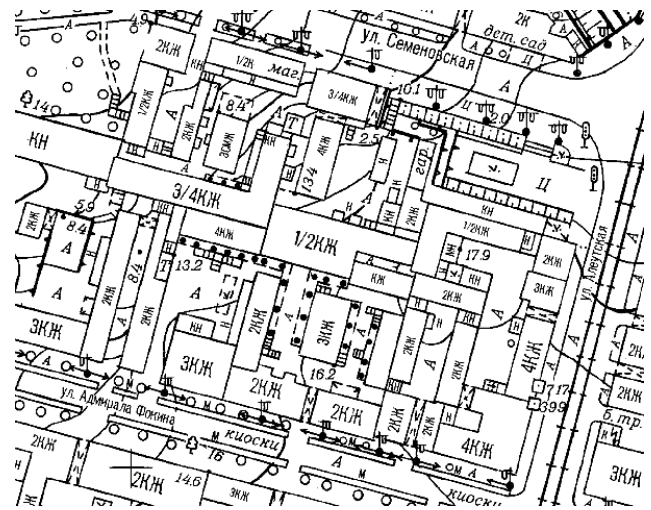


Рис. 5: Топографический план №1.

Полученные результаты позволяют говорить об эффективности метода. Даже при малом количестве эталонов в базе, количество верно распознанной информации больше 70%. Трудностью для алгоритма является определение таких символов, как «6» и «9», «К» и «Н». Однако, правило записи

информации на топоплане позволяет легко уточнить полученный результат.

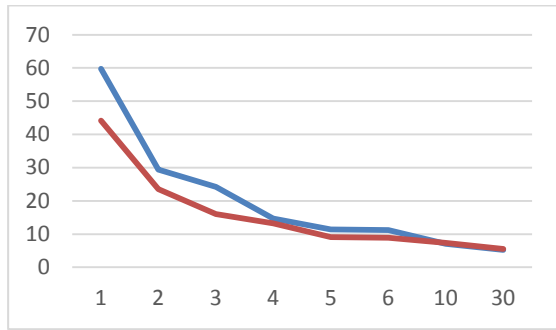


Рис. 6:Время загрузки базы эталонов (синяя линия) и время распознавания символов (красная линия). Подпись снизу – угол поворота (в градусах), подпись справа – время (в секундах)

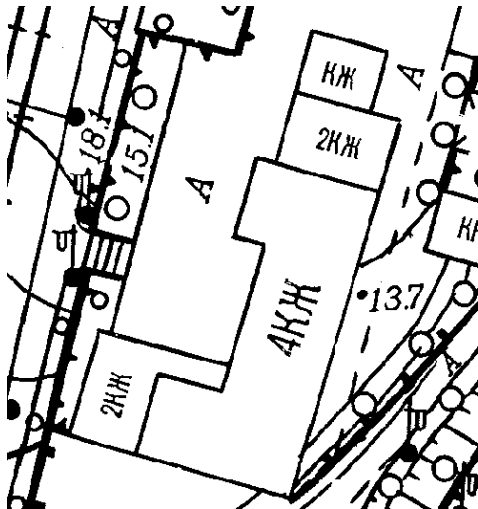


Рис. 7: Топографический план №2.

Предложенный алгоритм был применён для реконструкции реальной сцены по топографическому плану, изображенного на рисунке 1. Время загрузки базы эталонов составило 26,2 секунды при $\alpha=2^\circ$. Время выделения контуров заняло менее секунды.

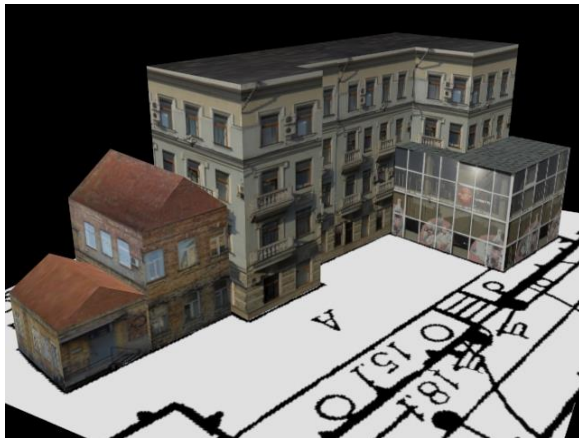


Рис. 8.а: Реконструкция реальной сцены. Вид №1.

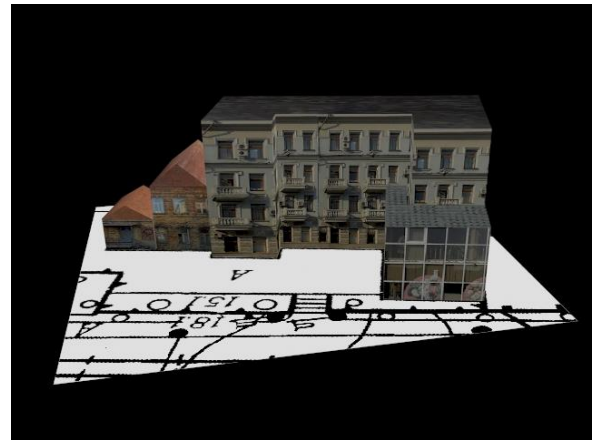


Рис. 8.б: Реконструкция реальной сцены. Вид №2.

Приведённые результаты были получены на машине со следующими характеристиками: IntelCorei3-2320VCPU 2200 GHz, 8 ГБ ОЗУ.

9. ЗАКЛЮЧЕНИЕ

Предложенные алгоритмы по выделению контуров показали свою эффективность, поэтому могут быть использованы для решения прикладных задач, связанных не только с реконструкцией, но и с другими областями компьютерного зрения.

Рассмотренные алгоритмы являются частью системы, задача которой состоит в реконструкции объектов городской обстановки по топографическому плану.

10. ССЫЛКИ

- [1] Lee, C.Y., «An Algorithm for Path Connections and Its Applications», IRE Transactions on Electronic Computers, vol. EC-10, number 2, pp. 364—365.
- [2] Бобков В.А., Соловьев И.В., Алгоритм выделения контуров зданий на топографических планах //Материалы Международного научного форума студентов, аспирантов и молодых ученых стран Азиатско-Тихоокеанского региона - 2012, Владивосток, 14-17 мая 2012 г. /под общей редакцией Н.В. Воеводиной. - Владивосток: Издательский дом Дальневост. федерал. ун-та, 2012. - С.- 62..
- [3] Кудряшов А.П. Реконструкция трехмерных сцен городской обстановки. Информационные технологии, №7, 2009. с. 63-68.

Обавторе

Соловьёв Игорь Владимирович – магистр по направлению прикладная математика и информатика. Окончил Дальневосточный федеральный университет в 2013 году. Его адрес igorek.solovyev@mail.ru.