

Направление вектора градиента совпадает с направлением максимальной скорости изменения функции в точке

$$\nabla f = \sqrt{G_x^2 + G_y^2}. \quad (8)$$

Определим угол между направлением вектора  $\nabla f$  в точке  $(x, y)$  и осью  $OX$ :

$$\alpha(x, y) = \arctg\left(\frac{G_y}{G_x}\right). \quad (9)$$

Отсюда можно легко найти направление контура в точке  $(x, y)$ , перпендикулярном направлению вектора градиента в этой точке.

Данную процедуру проводим отдельно для двух матриц  $[WX'_{i,j}]$  и  $[WY'_{i,j}]$ , а полученный контурный препарат складываем по схеме логического «ИЛИ» (см. блок 25, рис. 6).

Известно, что качество сегментации зависит во многом от вида изображения. Поэтому рекомендуемый выше метод Канны может дать отличные результаты для одного класса изображений, а для другого – плохие. В таком случае возникает необходимость в адаптации метода Канны для сегментации цветных изображений дефектов на панелях солнечных батарей. Поскольку неразрушающий контроль панелей должен осуществляться в реальном масштабе времени, возникает вторая актуальная задача, связанная с повышением производительности модернизированного алгоритма Канны, описанного выше.

**Заключение.** В статье выполнен анализ требований к камерам средств видеонаблюдения, дефектов фольгового материала солнечных панелей в процессе их тестирования. Описан метод и разработан алгоритм выделения контуров изображений при тестирования солнечных панелей, основанный на методе Канны для сегментации изображений дефектов серого цвета.

Упомянуто перспективное направление дальнейших исследований, связанных с адаптацией метода Канны для сегментации цветных изображений дефектов на панелях солнечных батарей.

#### СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Kurtz, S. Qualification Testing Versus Quantitative Reliability Testing of PV / S. Kurtz, K. Whitfield, N. Phillips, T. Sample, C. Monokroussos, E. Hsi, I. Repins, P. Hacke, D. Jordan, J. Wohlgemuth, P. Seidel, U. Jahn, M. Kempe, T. Tanahashi, Y. Chen, B. Jaeckel, M. Yamamichi // *Gaining Confidence in a Rapidly Changing Technology*: Preprint. United States, 2017. – 12 p.
2. Davis, K. O. Manufacturing metrology for c-Si module reliability and durability / K. O. Davis, M. P. Rodgers, G. Scardera, R. P. Brooker, H. Seigneur, N. Mohajeri, N. G. Dhare, J. Wohlgemuth, E. Schneller, N. Shiradkar, A. C. Rudack, W. V. Schoenfeld 85 // *Renewable and*

*Sustainable Energy Reviews*. – Vol. 59. – P. 225–252, June 2016. – Part II: Cell manufacturing NREL/JA-5J00-658.

3. Ez-Zahout, A. A framework for big data analytics in secure network of video surveillance systems based on images indexation / A. Ez-Zahout, J. Oubaha // *Proceedings of the 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, 2017. – P. 1–5.
4. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – Москва : Техносфера, 2005. – 1072 с.
5. Прэт, У. Цифровая обработка изображений: в 2-х книгах / Пер. с англ. Д. С. Лебедев. – М.: Мир, 1982. – Кн. 2. – 480 с.
6. Полякова, М. В. Морфологический метод контурной сегментации на основе репагулярного вейвлет-преобразования / М. В. Полякова, В. Н. Крылов // *Труды Одесского политехнического университета*. – 2006. – Вып. 1 (25). – С. 98–103.
7. Zahorodnia, D. Structural statistic method identifying facial images by contour characteristic points / D. Zahorodnia, Y. Pigovsky, P. Bykovyy, V. Krylov, I. Paliy, I. Dobrotvor // *Proceedings of the 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS'2015)*, September 24–26, Poland, Warsaw, 2015. – P. 293–297.
8. Zahorodnia, D. Canny-based method of image contour segmentation / D. Zahorodnia, Y. Pigovsky, P. Bykovyy // *International Journal of Computing*. – Vol. 15. – Issue 3. – P. 200–205, 2016. – Retrieved from <http://www.computingonline.net/computing/article/view/853>
9. Marr, P. Robust regression for computer vision: a review / P. Marr, M. Doron // *International Journal of Computer Vision*. – Vol. 6. – Issue 1. – 1991. – P. 59–70.
10. Ramadevi, Y. Segmentation and object recognition using edge detection techniques // *International Journal of Computer Science and Information Technology*. – Vol 2. – № 6. – 2010 – P. 153–161.
11. Senthikumaran, N. Edge detection techniques for image segmentation – a survey of soft computing approaches / N. Senthikumaran, R. Rajesh // *International Journal of Recent Trends in Engineering*. – Vol. 1. – № 2. – 2009. – P. 250–254.
12. Rosenfeld, A. Image segmentation and image models / A. Rosenfeld, L.S. Davis // *Proceedings of IEEE*. – Vol. 67. – Issue 5. – 1979. – P. 764–772.
13. Canny, J.E. A computational approach to edge detection / J. E. Canny // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – № 8. – 1986. – P. 679–698.
14. Krylov, V. The method of image contour segmentation based on wavelet transform and mathematical morphology / V. Krylov, M. Polyakova // *Proceedings of the International Conference on Modern Problems of Radioengineering, Telecommunications and Computer Science: TCSET'2006*. – Lviv, 2006. – P. 236–239.

Материал поступил в редакцию 21.02.2019

#### GRZESZCZYK KONRAD Images processing during the testing of solar panels

The article presents the modification of the Canny method for the contour segmentation of images. The peculiarity of the modified method and a corresponding developed algorithm is the use of wavelet functions for the allocation of co-structures when converting an image. That enabled to obtain a sequence of contours of the hierarchical object with the adjustable details. The proposed approach is aimed for detecting the contours of noise images defects at the foil material of solar panels.

УДК 681.3

Матюшкова Г. Л., Матюшков А. Л., Войцехович О. Ю.

### АЛГОРИТМ ОТЫСКАНИЯ КРАТЧАЙШЕГО ПУТИ ОТ ЗАДАННОЙ ВЕРШИНЫ ГРАФА ДО ОСТАЛЬНЫХ

**Введение.** Многие исследователи мира только в последних двух столетиях разработали сотни алгоритмов поиска различных путей как между двумя заданными вершинами графа, так и одной или несколь-

кими и в десятках других вариантов в зависимости от конфигурации графа или класса графов, которые использовались за основу описания реального явления, что послужило почвой для разработки специ-

Войцехович Оксана Юрьевна, ст. преподаватель кафедры ИИТ Брестского государственного технического университета. Беларусь, 224017, г. Брест, ул. Московская, 267.

Физика, математика, информатика

фических узконаправленных алгоритмов. Многие из них и основные термины приводятся в книге Зыкова [1]. Второй важной причиной к созданию большого числа разных алгоритмов стало использование вычислительной техники, что потребовало специальной организации данных, вычислительного процесса и шагов алгоритмов, направленных на экономию памяти ЭВМ и быстроедействие алгоритма. Базовым считается алгоритм Дейкстры [2], в котором развивался волновой подход при отыскании кратчайшего пути на графе от заданной вершины  $i$  до искомой  $j$ , когда граф связан и длина его связей  $d_{ij}$  больше или равна нулю. Однако в ряде приложений желательно найти не только самую длину пути, но и номера вершин через которые он проходит, а иногда и альтернативные пути такой же длины.

**Алгоритм поиска кратчайшего пути от заданной вершины графа до всех остальных.** Для правильного использования алгоритма перечислим применяемые в нём понятия и соглашения. Термин «волновое построение простых цепочек между вершинами графа» трактуется как их последовательное получение от текущей вершины по всем возможным рёбрам за один шаг (кроме рёбер, приводящих к образованию цикла, т. е. запрещается строить цепочку с повторением уже использованных номеров в ней), и далее аналогичная операция повторяется для всех вершин, которые участвуют в образовании цепочек такого же порядка, т. е. с таким же количеством номеров вершин. При исчерпании всех цепочек данной длины берется последняя вершина в первой цепочке предыдущей длины (порядка) и для неё строятся все цепочки следующего порядка и так поступают со всеми цепочками предыдущего порядка до их исчерпания. Этот процесс напоминает волновой, когда круги от брошенного камня последовательно распространяются по воде. Исходной вершине присвоим номер  $k$  (часто для удобства принимают  $k=1$ ). Расстояние  $d_{kk}$  принимается равным 0, так как в вершину от самой себя к ней же перемещать не нужно. Отсутствие связи между вершинами  $i$  и  $j$  обозначим – (черточкой), а иногда для этой цели ставят знак бесконечность. Любую цепочку будем записывать в виде  $k, i_1, i_2, \dots, i_r - S_{k, i_r}^*$ , где  $k$  – номер начальной вершины, номера  $i_r$  остальных вершин в порядке формирования этой цепочки, – (разделитель цепочки от значения её длины, \* – знак конца данной записи).

Основные элементы алгоритма представим в виде следующих этапов:

1. Задание исходной информации в виде матрицы  $n \times n$  ( $n$  – число вершин графа). Элементами матрицы будут в строке  $i$  числа, соответствующие расстоянию от вершины  $i$  до её соседи  $j$  в порядке расположения столбцов матрицы, в рассматриваемой строке (например, на пересечении строки 3 и столбца 5 это будет расстояние от вершины графа 3 до 5. Для удобства вычислений расстояния типа  $d_{kk}$  помечаются нулём, а отсутствующие связности черточкой (-), чтобы не строить заведомо не существующие пути (см. пример). Задаётся расстояние  $S_{kk}=0$  от  $k$  до  $k$

2. Записывается вырожденная первая цепочка из одного числа  $k$  и ставится  $-0^*$ :

$$k-0^*$$

3. Строятся все допустимые простые цепочки 2, 3...и далее порядков по структуре:

$$k, i_1 - S_{k, i_1}^* \quad k, i_1, i_2 - S_{k, i_2}^* \quad \dots \quad k, i_1, i_2, i_3, \dots, i_r - S_{k, i_r}^* -$$

(общая структура любой цепочки)

$$k, i_1, i_2 - S_{k, i_2}^*$$

$$k, i_1, i_2, i_3 - S_{k, i_3}^*$$

$$k, i_1, i_2, i_3, \dots, i_m - S_{k, i_m}^*$$

4. Исключаются все лишние цепочки для путей, ведущих в каждую из вершин  $1, 2, \dots, n$ , т. е. остаются лишь те, у которых для последнего данного номера  $i_v$  цепочки с  $S_{k, i_v}$  – минимальны (при наличии нескольких путей одной длины их можно запоминать с целью построения резервных путей или альтернативных решений).

5. Печать номеров вершин и расстояний до них от указанной в задании вершины  $k$ , а также номеров вершин, через которые проходят пути такой же длины.

Таким образом, получаются все расстояния до всех вершин от заданной и перечни номеров вершин, через которые они проходят.

Пример вычислений по алгоритму

1. Задание матрицы ( $6 \times 6$ ) и начальной вершины 1.

0	7	9	-	-	14	1)
7	0	10	15	-	-	2)
9	10	0	11	-	2	3) (этот столбец с номерами строк матрицы для поиска $S_{ij}$ )
-	15	11	0	6	-	4)
-	-	-	6	0	9	5)

- 14 - 2 - - 9 0 6)

2. Запись цепочки первого порядка 1-0\*
3. Записи остальных цепочек от 2 до  $m < n$  порядков ( $m=n$  случай, когда имеются цепочки, проходящие через все вершины. Как правило, в графах с ограниченным числом связей  $u$  вершин их не бывает):

$$1,2-7^*$$

$$1,3-9^*$$

$$1,6-14^*$$

$$1,2,3-17^*$$

$$1,3,2-19^*$$

$$1,3,4-20^*$$

$$1,3,6-11^*$$

$$1,3,4,2-35^*$$

$$1,3,4,5-26^*$$

$$1,3,6,5-20^*$$

4. Исключение лишних цепочек: среди цепочек второго порядка нужно изъять (1,6-14\*), так как вершина 6 достигается за 11 единиц в цепочке (1,3,6-11\*) следующего порядка (по аналогичным причинам изымаются цепочки (1,2,3-17\*) и (1,3, 2-19\*), так как среди цепочек 2-го порядка имеются (1,2-7\*) и (1,3-9\*) с лучшими показателями. Для цепочек четвёртого порядка (1,3,4,2-35\*) и (1,3,4,5-26\*) находим цепочки (1,2-7\*) и (1,3,6,5-20\*) с меньшими показателями соответственно.

При исключении цепочек по сути дела надо сравнить только все те, которые имеют одинаковую последнюю вершину, и оставить цепочки с одинаковым кратчайшим путём.

5. Печать оставшихся цепочек: 1,2-7\*; 1,3-9\*; 1,3,4-20\*; 1,3,6-11\*; 1,3,6,5-20\*.

Для контроля результатов можно использовать следующий граф, представленный на рисунке 1, который соответствует нашей матрице.

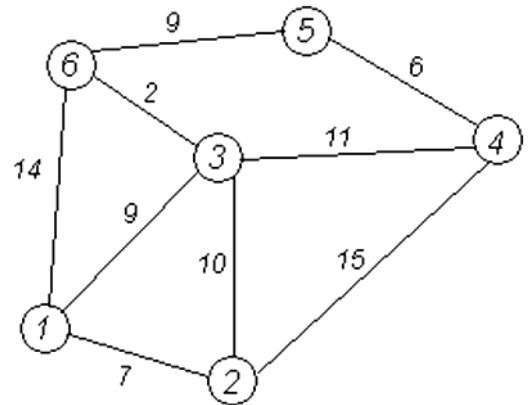


Рисунок 1 – Граф для проверки результатов

На этом графе (рис. 1) можно визуально убедиться в правильной работе алгоритма. Покажем это на примере цепочек от вершины 1 до вершин 4 и 5. Алгоритм дал два пути с длиной 20 единиц (1,3,4 и

1,3,6,5). Сразу легко проверить, что они существуют и их длина 20 единиц. Внимательно изучив граф, находим ещё пути из 1 в 5 и из 1 в 4, убеждаемся, что они больше.

**Заключение.** В статье предложен алгоритм с умеренным перебором вариантов, который для графов с ограничениями на количество связей между вершинами и учёте особенностей моделируемого объекта на их характер обеспечивает следующие преимущества:

- 1) для всех заданных пар вершин графа называются кратчайшие пути и перечень вершин, через которые они проходят;
- 2) не требуется особых отметок для вершин, которые уже рассмотрены;
- 3) возможно указание всех цепочек с одинаковым расстоянием от заданной вершины;

4) не требуется в виде какого-то числа называть машинную бесконечность для условий конкретного графа;

5) исключение цепочек с путём более кратчайшего делается всего лишь по равенству последних номеров их вершин при совмещении с операцией сравнения их путей;

6) естественным путём заканчивается процесс построения цепочек. Следует отметить, что иногда из-за равнозначных ответов их количество может превышать число вершин заданного графа.

#### СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. Зыков, А. А. Основы теории графов. – М.: Наука, 1987.
2. Алгоритм Дейкстры / Википедия. – Дата доступа: 10.11.2018.

Материал поступил в редакцию 08.02.2019

**MATYUSHKOVA G.L., MATYUSHKOV A. L., VOYTSEKHOVICH O. Yu. Algorithm of search of the shortest way from the set count's top to the others**

The article deals with the wave algorithm for finding the shortest path between the given vertices of the graph with a list of vertices on which it passes.

УДК 519.673

**Петренко Т. Ю., Петренко И. А., Дереченник С. С.**

## О МЕТОДЕ ФОРМИРОВАНИЯ ДЛИННОПЕРИОДНОЙ КОНГРУЭНТНОЙ ПСЕВДОСЛУЧАЙНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ

**Введение.** Псевдослучайные последовательности (ПСП) чисел имеют широкое применение в разных приложениях – от метода Монте Карло и имитационного моделирования до криптографии. Существует достаточное количество требований качества, предъявляемых к генераторам псевдослучайных последовательностей. Одним из этих требований является достаточно длинный период, гарантирующий отсутствие зацикливания последовательности в рамках решения поставленной проблемы. Задача формирования длиннопериодных ПСП является актуальной, поскольку от длины периода используемых ПСП напрямую зависит качество результатов. Известно, что ни одна псевдослучайная последовательность не может быть абсолютно случайной, поэтому последовательности должны проходить статистическую проверку. В данной статье впервые представлен алгоритм и анализ модифицированного конгруэнтного метода формирования псевдослучайных чисел, который способен формировать длиннопериодные псевдослучайные последовательности.

**Линейный конгруэнтный метод.** В данном давно изученном методе выбираются, как описывал Д. Кнут [1], четыре «волшебных числа»:

$m$ , модуль;  $0 < m$ ;

$a$ , множитель;  $0 \leq a < m$ ;

$c$ , приращение;  $0 \leq c < m$ ;

$X_0$ , начальное значение;  $0 \leq X < m$ .

Получают линейную конгруэнтную последовательность по формуле 1.

$$X_{n+1} = (aX_n + c) \bmod m, n \geq 0. \quad (1)$$

Нам интересен случай с принятым названием «мультипликативный» линейный конгруэнтный метод, т. е. при  $c = 0$ . Обычно модуль выбирается равным длине машинного слова  $w = 2^e$ . Д. Кнут описал, что суммирование по модулю  $w$  (кроме случаев, когда машина использует процедуру единичного дополнения) и умножение по модулю  $w$  простое, т. к. затрагиваются только младшие разряды произведения. Существует случай, при  $c = 0$ , когда  $m = w + 1$ , т. е.

значение  $aX_n$  может находиться между 0 и  $w$  включительно. Объясняется это тем, что переполнение происходит только тогда, когда результат равен  $w$ , и удобно его отбросить, когда оно появляется в конгруэнтной последовательности по модулю  $m = w + 1$ . В таком случае младшие разряды  $X_n$  ведут себя так же случайно, как и старшие. При этом важно учесть свойства коэффициентов, которые для получения максимального периода следующие:

1) числа  $c$  и  $m$  взаимно простые;

2)  $b$  кратно  $p$  для каждого простого  $p$ , являющегося делителем  $m$ ;

3) если  $m$  кратно 4, то и  $b$  должно быть кратно 4.

По условиям, описанным еще в 1990 году [2], для получения случайных характеристик множитель  $a$  при делении на 8 должен давать остаток 5, т. е.  $a \equiv 5 \pmod{8}$ . Начальное число  $X_0$ , задаваемое оператором, для получения хороших статистических характеристик, должно давать при делении на 4 остаток 1, т. е.  $X_0 \equiv 1 \pmod{4}$ . Для того чтобы не заниматься подбором этих двух чисел с условиями, воспользуемся формулами  $a = 5^{2^{p+1}}$ , а начальное число заменим на  $RX_0 = 4X_0 + 1$  (тогда любое число, которое мы запишем в  $X_0$ , запишет в  $RX_0$  число, которое при делении на 4 дает остаток 1). Рассмотрим модуль  $m$ , для заданного значения множителя  $m = 2^N$ , где  $N$  – это разрядность двоичного числа, т.е. количество числовых разрядов, необходимых для записи этого числа в двоичной системе счисления. Для определения степени  $2p + 1$  должно выполняться условие  $5^{2^{p+1}} < 2^N$ , т. е.

$a < m$ . Прологарифмировав левую и правую части неравенства, найдем наибольшее значение  $5^{2^{p+1}} = \text{fix}[N \ln 2 / \ln 5]$ , где операция  $\text{fix}[5.65] = 5$  оставляет только целую часть числа. Период

**Петренко Татьяна Юрьевна**, аспирант кафедры электронных вычислительных машин и систем Брестского государственного технического университета.

**Дереченник Станислав Станиславович**, к. т. н., доцент, зав. кафедрой электронных вычислительных машин и систем Брестского государственного технического университета.

Беларусь, БрГТУ, 224017, г. Брест, ул. Московская, 267.

**Петренко Илья Александрович**, студент кафедры прикладной информатики и математики Варшавской главной школы сельского хозяйства.

Poland, 02-787 Warszawa, ul. Nowoursynowska, 166.