

# Алгоритм волновой скелетизации растровых изображений

М.А. Кудрина<sup>1</sup>, В.С. Мишенев<sup>1</sup>

<sup>1</sup>Самарский национальный исследовательский университет им. академика С.П. Королева, Московское шоссе 34А, Самара, Россия, 443086

**Аннотация.** В статье приведено описание волнового алгоритма скелетизации растрового изображения. Представлены схемы алгоритма построения первичного графа и алгоритма разделения генерации волны на дочерние волны. Для хранения структуры графа предложено использовать гнездовую структуру, что позволяет экономить память и легче изменять структуру графа. Для первичной оптимизации графа в методе последовательных приближений предложено использовать ортогональную регрессию Деминга.

## 1. Введение

Существует ряд задач, в которых необходимо построить скелет растрового изображения, к примеру, распознавание рукописного текста, сравнение отпечатков пальцев, обработка медицинских, картографических изображений, технических чертежей [1-5] и т.п.

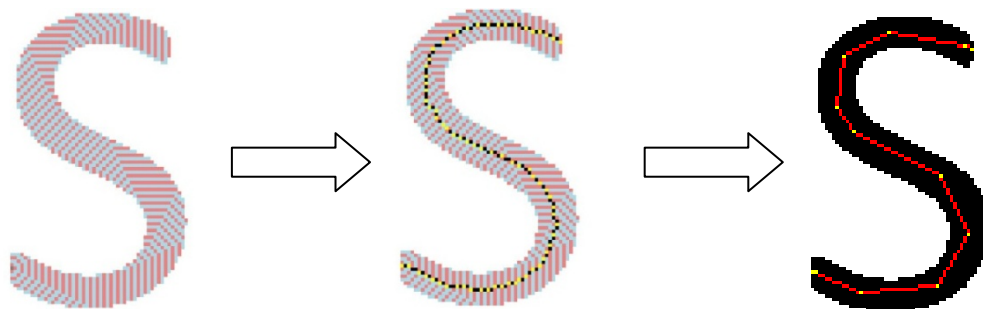
Существуют различные алгоритмы скелетизации: алгоритм Зонга-Суэна (Zhang-Suen), шаблонный метод, волновой алгоритм [1, 6, 7] и др. В данной статье рассматривается волновой алгоритм.

Основная суть волнового алгоритма для нахождения скелета растрового изображения хорошо описана и иллюстрирована, в частности, в работах [1, 4, 6]. Однако при реализации программ с использованием данного метода может возникнуть ряд вопросов, тонкостей и нюансов, раскрытие которых и стало предметом написания данной статьи. В статье приведены схемы алгоритма построения первичного графа и алгоритма разделения генерации волны на дочерние волны. Научная новизна работы заключается в использовании гнездовой структуры для хранения графа, описывающего скелет объекта, вместо матриц смежности и инцидентности, что позволяет экономить память и легче менять структуру графа, а также в использовании ортогональной регрессии Деминга в методе последовательных приближений для первичной оптимизации графа.

Скелетом изображения будем называть множество точек, равноудаленных от границ изображения. Этапу построения скелета фигуры, как правило, предшествуют несколько вспомогательных этапов. Это предварительная обработка (устранение мелких шумов и пр.) и бинаризация изображения. Бинаризацией называется процесс преобразования цветных и полутоновых изображений в двухцветные, т.е. бинарные. Для бинаризации используются пороговая обработка, методы точечных преобразований, свертки, усиления краев, выделения низкочастотных и высокочастотных компонент изображения и т. д.

Алгоритм волновой скелетизации условно можно разбить на два основных этапа: этап построения первичного графа (включает в себя запуск волны, отслеживание пройденного пути, мест разделения и затухания волны) и следующий за ним этап оптимизации графа, в результате чего отбрасывается лишняя информация. Иллюстрация работы данных этапов приведена на рисунке 1.

На первом этапе внутри объекта запускается сферическая волна. Генерация волны – это фронт волны. Точки, принадлежащие отдельным генерациям волны, помечаются разным цветом на изображении, и отслеживаются центральные пиксели каждой нечетной генерации, которые помещаются в первичный граф как его узлы.



**Рисунок 1.** Основные этапы скелетизации: построение первичного графа и оптимизация графа.

Первичный граф, однако, обычно содержит большое количество избыточной информации, поэтому на этапе оптимизации графа лишние точки из него удаляются.

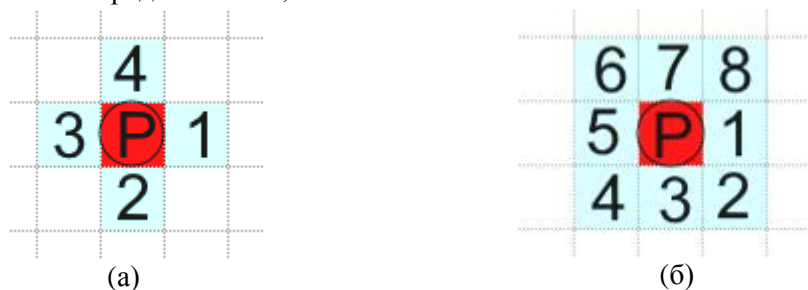
## 2. Этап построения первичного графа

Построение графа осуществляется путем отслеживания пути прохождения сферической волны по изображению.

### 2.1. Распространение сферической волны

Сферическая волна запускается из любого пикселя внутри объекта.

Для получения генерации сферической волны применяется попеременно 4-х и 8-ми связное распространение. Связность характеризует количество соседних пикселей для текущего пикселя. Для 4-х связного распространения волны соседними считаются верхний, нижний, правый и левый пиксели текущего пикселя (см. рисунок 2(а)). Для 8-ми связного распространения, соответственно, соседними считаются все 8 пикселей вокруг текущего (см. рисунок 2(б)). На рисунке 2 буквой Р обозначен текущий пиксель, а пронумерованы его соседи для 4-х и 8-ми связных представлений, соответственно.



**Рисунок 2.** Соседние пиксели и порядок их обхода для 4-х связного (а) и 8-ми связного (б) представлений.

При запуске волны сначала применяется 4-х связное распространение, затем 8-ми связное для каждого пикселя из предыдущей 4-х связной генерации и так далее поочередно. Тогда

распространение волны идёт в виде восьмиугольника. В основе данного подхода лежит принцип Гюйгенса-Френеля волновой теории, согласно которому каждый элемент волнового фронта можно рассматривать как центр вторичного возмущения, порождающего вторичные сферические волны.

Для сферической волны характерны некоторые особенности:

- не более чем через  $2N$  шагов распространение волны приобретает устойчивый характер, независимо от начальной точки распространения волны, где  $N$  – ширина линии в пикселях (см. рисунок 3);
- такая волна «умеет» поворачивать, хорошо огибает различные помехи. Небольшие помехи в 1-2 пикселя мало влияют на распространение волны. Однако такие помехи лучше удалять на этапе получения бинарного изображения для устойчивости волны.

Нулевая генерация состоит из одного начального пикселя. Он помечается на изображении первым. На рисунке 4 он расположен в центре.

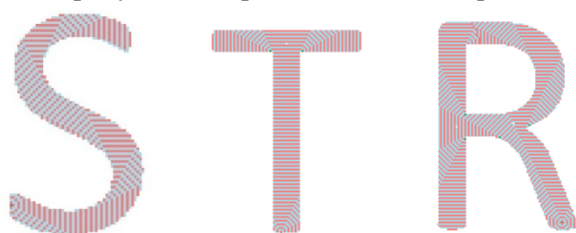


Рисунок 3. Распространение сферической волны.

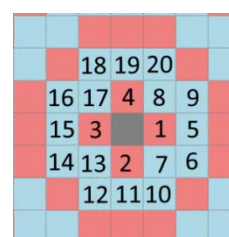


Рисунок 4. Нумерация пикселей первой и второй генерации.

Первая генерация волны состоит из непомеченных соседей начального пикселя при 4-х связном распространении. Пиксели первой генерации помечаются на изображении красным цветом. На рисунке 4 они имеют номера 1, 2, 3, 4. Вторая генерация – это непомеченные соседи пикселей первой генерации при 8-ми связном распространении (на рисунке 3 имеют номера с 5 по 20) и т.д.

Для каждой точки с координатами  $(i, j)$  проверка, помечены ли её соседние пиксели, происходит в фиксированном порядке - всегда начиная с правого среднего пикселя по часовой стрелке. Тогда при 4-х связном распространении проверка происходит в порядке, изображенном на рисунке 2(а), а при 8-ми связном распространении – как на рисунке 2(б). На рисунке 4 можно заметить, что порядок обхода пикселей второй генерации в примере не соответствует порядку пикселей на изображении. Это происходит из-за фиксированного порядка пометки соседних пикселей, однако это упрощает алгоритм.

## 2.2. Структура графа

Граф объекта хранится в памяти в виде гнездовой структуры [8]. При данном способе хранения структуры графа каждый его узел задается гнездом, состоящим из координат узла графа  $(x, y)$  и списка адресов переходов к следующим (дочерним) гнездам (узлам графа). Этот способ удобен тем, что структуру графа легко менять и не требуется большого объема памяти, в отличие от матриц смежности или инцидентности.

На рисунке 5 изображен пример графа и соответствующая ему гнездовая структура. Гнезда содержат координаты узла графа  $(x_i, y_i)$  и  $A_i$  – ссылку на список, содержащий ссылки на соседние гнезда (узлы графа).

## 2.3. Описание основных алгоритмов построения первичного графа

Схема алгоритма построения первичного графа представлена на рисунке 6.

Построение скелета фигуры производится путем отслеживания центров каждой нечетной генерации волны. Текущая генерация волны хранится в памяти в виде списка координат пикселей данного фронта. Список точек любой нечетной генерации обозначим  $S_n$ , поскольку точки, входящие в нее, являются 4-х связными соседями точек предыдущей генерации.

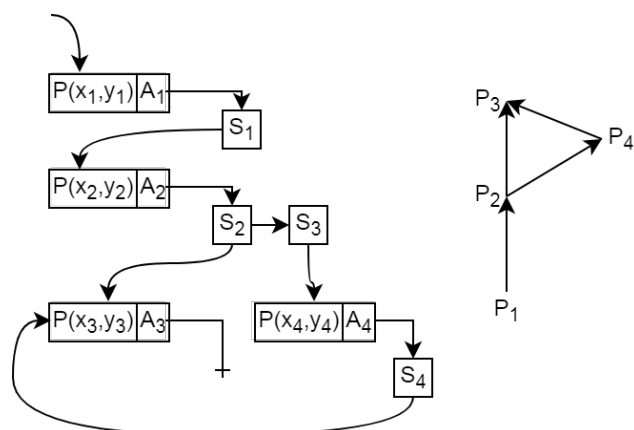


Рисунок 5. Гнездовая структура для хранения графа объекта.

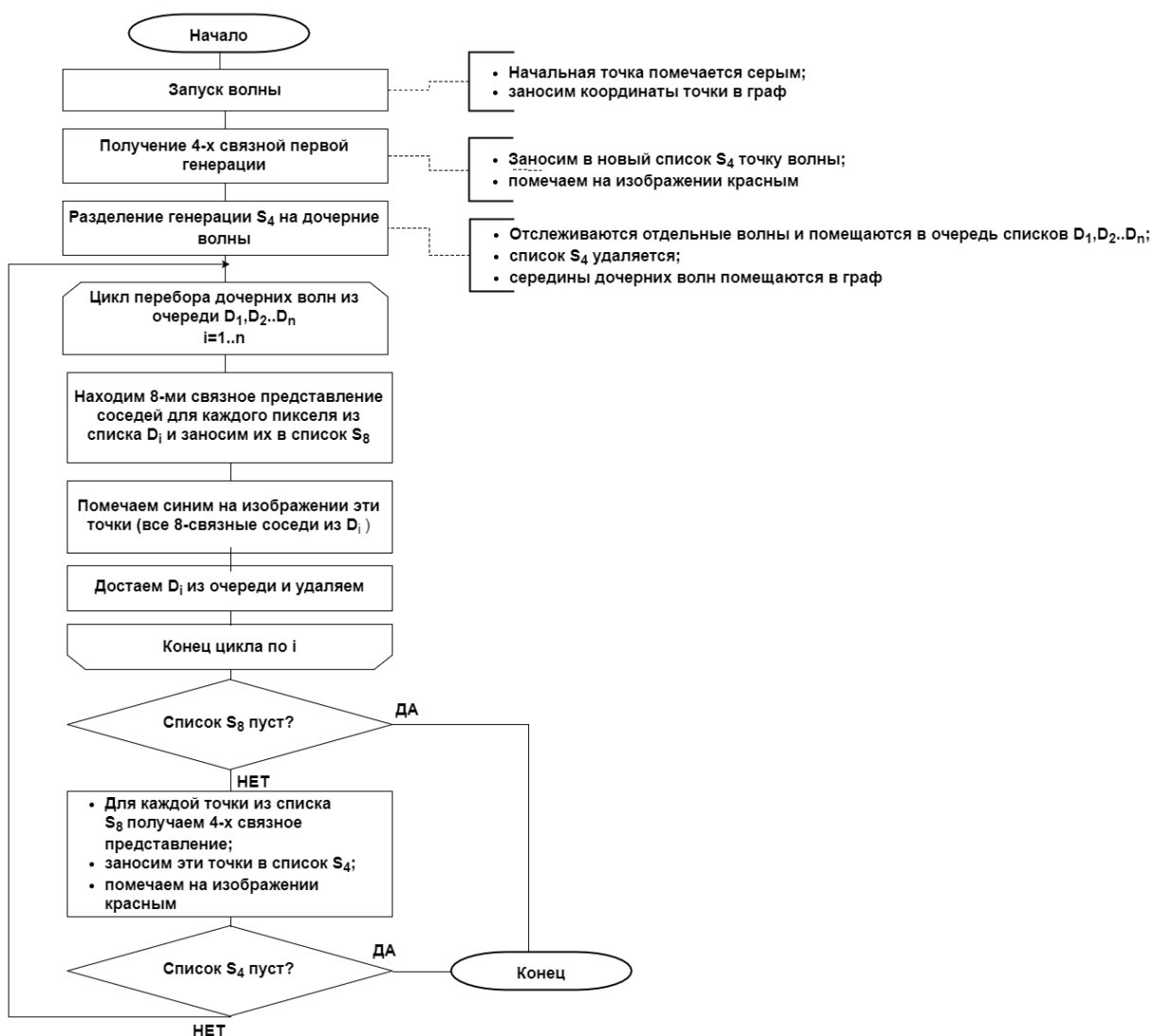


Рисунок 6. Схема алгоритма построения первичного графа.

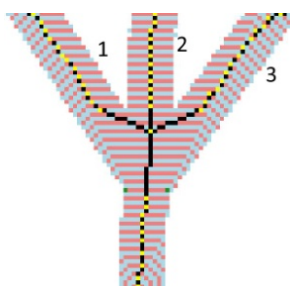
Соответственно, списки четных генераций обозначим  $S_8$ , т.к. они являются 8-ми связными соседями точек предыдущей генерации.

Каждая генерация волны заключена между своими крайними пикселями с координатами  $(x_1, y_1)$  и  $(x_2, y_2)$ . Под крайними пикселями волны понимаются те пиксели генерации, у которых

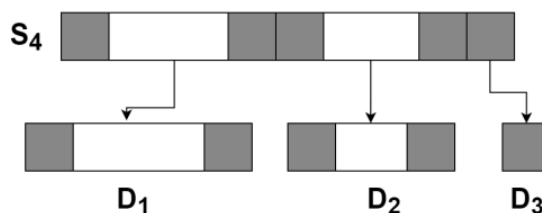
не более одного соседнего пикселя из этой генерации. За точку скелета фигуры будем брать середину отрезка, соединяющего крайние точки генерации волны, т.е. координаты точки будут определяться как  $((x_1 + x_2)/2; (y_1 + y_2)/2)$ .

Кроме того, возможна ситуация, когда генерация волны состоит из нескольких отдельных фронтов. Такое происходит при разделении генерации волны на дочерние волны.

Данную ситуацию можно отследить по количеству крайних точек в списке  $S_4$  точек генерации. Если количество крайних точек больше двух, то произошло разделение, и список содержит несколько дочерних волн (на рисунке 7 приведен пример разделения волны на 3 фронта). В этом случае отдельные списки точек дочерних волн ( $D_1, D_2, \dots, D_n$ ) помещаются в очередь. На рисунке 8 приведена иллюстрация разделения списка генерации волны  $S_4$  на дочерние волны  $D_1, D_2, \dots, D_n$ .



**Рисунок 7.** Разделение генерации волны на дочерние волны.



**Рисунок 8.** Список точек генерации волны  $S_4$  разделяется на 3 фронта.

Схема алгоритма разделения генерации волны ( $S_4$ ) на дочерние волны ( $D_1, D_2, \dots, D_n$ ) приведена на рисунке 9.

Перед разделением наблюдается непрерывное увеличение ширины волны и происходит увеличение количества крайних точек генерации волны. Отслеживая это, можно определить место соединения отрезков скелета изображения. Место соединения отдельных веток находится как центр масс многоугольника, образованного крайними точками материнской и дочерних волн [5, 6, 9].

В случае поворота волны или утолщения элемента изображения количество крайних точек не изменяется, поэтому такое увеличение ширины волны можно не принимать во внимание.

При встрече с другой волной происходит затухание волны. В данном случае граф замыкается.

### 3. Этап оптимизации графа

В начале работы для оптимизации графа (т.е. удаления избыточной информации) используется метод последовательных приближений, который заключается в следующем: через первые две вершины графа проводится прямая, и каждая следующая точка проверяется на отклонение от этой прямой. Если отклонение не больше  $\epsilon$  (задается соизмеримым с толщиной линий изображения), то эта точка принадлежит текущей прямой, иначе – образует следующую прямую.

Коэффициенты уравнения прямой  $Ax + By + C = 0$ , проходящей через две точки  $(x_1, y_1)$  и  $(x_2, y_2)$ , в методе последовательных приближений вычисляются по следующим формулам:

$$A = y_1 - y_2, B = x_2 - x_1, C = x_1 y_2 - y_1 x_2.$$

Тогда для каждой последующей точки  $(x_i, y_i)$  из графа находится расстояние от нее до прямой  $Ax + By + C = 0$  по формуле  $r = |Ax_i + By_i + C| / \sqrt{A^2 + B^2}$ . Если  $r < \epsilon$ , то эта точка принадлежит текущей прямой.

Полученный набор точек, принадлежащих одной прямой, аппроксимируется с помощью ортогональной регрессии. Ортогональная регрессия отличается от простой линейной регрессии в том, что она принимает во внимание ошибки в наблюдениях как по оси  $x$ , так и по оси  $y$  [10] (см. рисунок 10).

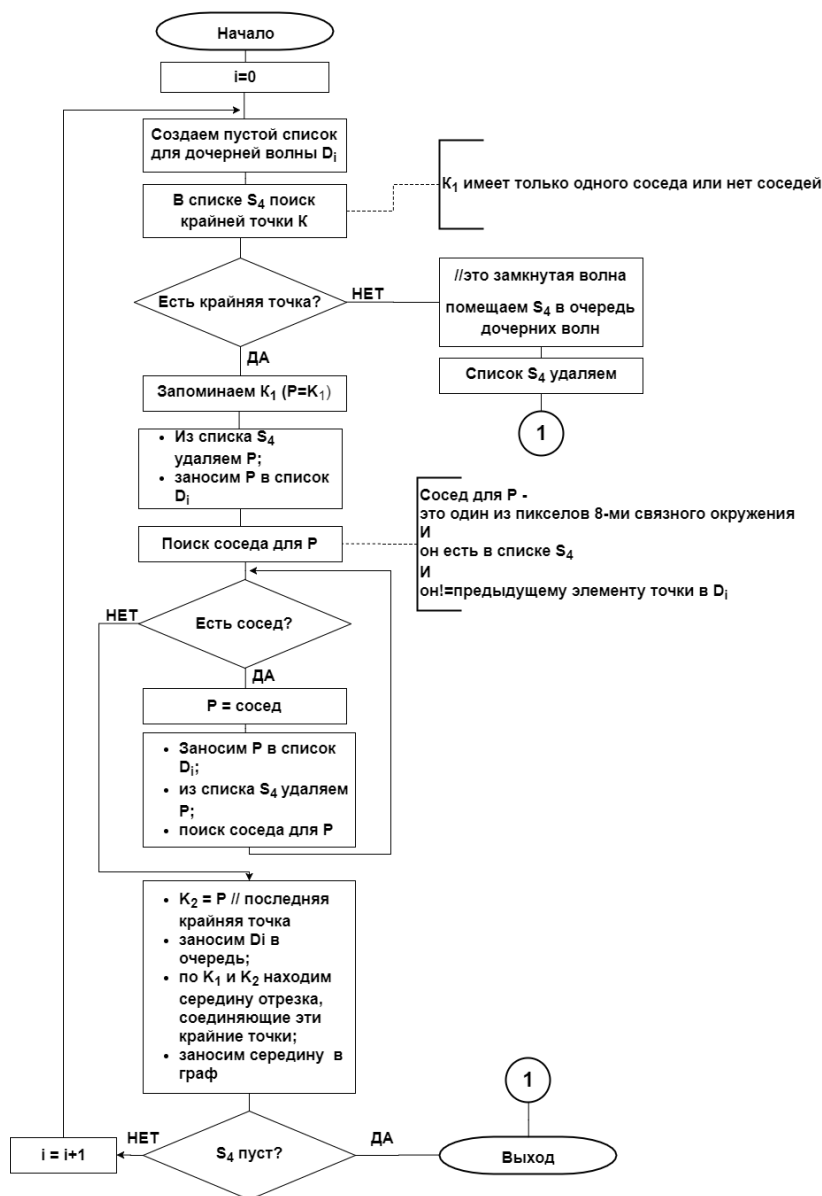


Рисунок 9. Схема алгоритма разделения генерации S4 на дочерние волны D1, D2, ..., Dn.

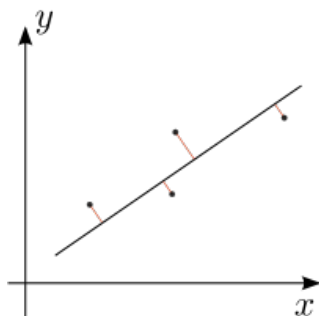


Рисунок 10. Аппроксимация набора узлов графа с помощью ортогональной регрессии Деминга.

Если точки аппроксимируются прямой, заданной уравнением  $y = \beta_0 + \beta_1 x$ , то коэффициенты регрессии вычисляются по следующим формулам [11]:

$$\beta_1 = (s_{yy} - s_{xx} + \sqrt{(s_{yy} - s_{xx})^2 + 4s_{xy}^2}) / 2s_{xy},$$

$$\beta_0 = \bar{y} - \beta_1 \bar{x},$$

где

$$\bar{x} = \frac{1}{n} \sum x_i, \bar{y} = \frac{1}{n} \sum y_i,$$

$$s_{xx} = \frac{1}{n-1} \sum (x_i - \bar{x})^2,$$

$$s_{yy} = \frac{1}{n-1} \sum (y_i - \bar{y})^2,$$

$$s_{xy} = \frac{1}{n-1} \sum (x_i - \bar{x})(y_i - \bar{y}).$$

Поскольку уравнение  $y = \beta_0 + \beta_1 x$  не описывает вертикальные прямые, приведем его к виду  $Ax + By + C = 0$ . Для этого вычислим коэффициенты  $A, B, C$ :  $A = \beta_1, B = -1, C = \beta_0$ .

Если  $s_{xx} = 0$ , то прямая параллельна оси  $Y$ , и уравнение имеет вид  $x = \bar{x}$ , коэффициенты прямой при этом:  $A = -1, B = 0, C = \bar{x}$ .

А если  $s_{yy} = 0$ , то прямая параллельна оси  $X$ , и уравнение имеет вид  $y = \bar{y}$ , коэффициенты прямой при этом:  $A = 0, B = -1, C = \bar{y}$ .

После нахождения коэффициентов аппроксимирующей прямой концевые точки корректируются, т.е. проецируются на аппроксимирующую прямую, а остальные промежуточные точки удаляются из графа.

Координаты  $(x_{proec}, y_{proec})$  проекции концевой точки  $(x_0, y_0)$  на прямую  $Ax + By + C = 0$ , полученную методом ортогональной регрессии, находятся по формулам:

$$x_{proec} = (B(Bx_0 - Ay_0) - AC)/(A^2 + B^2),$$

$$y_{proec} = (A(Ay_0 - Bx_0) - BC)/(A^2 + B^2).$$

Все пройденные вершины помечаются, во избежание заикливания программы, так как граф в общем виде может быть замкнутым.

#### 4. Результаты и выводы

В статье предложены схемы алгоритма построения первичного графа и алгоритма разделения генерации волны на дочерние волны. Для хранения структуры графа представления скелета объекта вместо матриц смежности и инцидентности предложено использовать гнездовую структуру, что позволяет экономить память и легче менять структуру графа. Научная новизна работы также заключается в использовании ортогональной регрессии Деминга в методе последовательных приближений для первичной оптимизации графа.

Разработана программа волновой скелетизации на языке C#. Данная программа предварительно преобразует цветное растровое изображение в бинарное путем пороговой обработки, где порог – среднее значение между максимальной и минимальной яркостью. После чего строится первичный граф и проводится его оптимизация.

Алгоритм волновой скелетизации не требует большого расхода памяти, поскольку память выделяется лишь для хранения графа скелета и списков пикселей текущей и следующей за ней генераций волн. Разметка сферических волн происходит на исходном изображении, исходное изображение легко можно восстановить. Алгоритм легко распараллеливается при использовании нескольких начальных точек запуска волны. Для запуска волны предпочтительно выбирать пиксели на нижних или верхних краях изображений символов, в этом случае каркасы получаются более ровными.

В настоящее время волновая скелетизация успешно используется для построения каркасов символов (букв, цифр и проч.), используемых для дальнейшей обработки при распознавании. Однако для создания, например, скелета отпечатков пальцев данный алгоритм подходит меньше, поскольку структура папиллярного узора пальцев сложнее, распространение волны и ее дальнейший анализ проходит с большими погрешностями [1].

#### 5. Литература

- [1] Гудков, В.Ю. Скелетизация бинарных изображений и выделение особых точек для распознавания отпечатков пальцев / В.Ю. Гудков, Д.А. Клюев // Вестник ЮУрГУ. Серия: Компьютерные технологии, управление, радиоэлектроника. – 2015. – №3. [Электронный

- ресурс]. – Режим доступа: <http://cyberleninka.ru/article/n/skeletizatsiya-binarnyh-izobrazheniy-i-vydelenie-osobyh-tochek-dlya-raspoznavaniya-otpechatkov-paltsev> (20.11.2017).
- [2] Погодин, С.В. Выделение и анализ скелетов объектов на цветных снимках / С. В. Погодин // Программные продукты и системы. – 2009. - № 2. – С. 42-45.
- [3] Кравченко, Ю. А. Представление и создание моделей неоднозначных топографических поверхностей / Ю. А. Кравченко // Геодезия и картография. – 2010. – № 7. – С. 11-15.
- [4] Каташевцев, М. Д. Волновая скелетизация / М. Д. Каташевцев // Вестн. Иркут. гос. техн. ун-та. – 2013. – № 7. – С. 89-92.
- [5] Романов, А.Н. Предварительная грубая скелетизация растровых изображений и детектирование простых элементов / А.Н. Романов, М.В. Привалов // Донбас-2020: перспективи розвитку очима молодих вчених: Матеріали V науково- практичної конференції. – Донецьк: ДонНТУ Міністерства освіти і науки. – 2010. – С. 518-522.
- [6] Клубков, И.М. Применение волнового алгоритма для нахождения скелета растрового изображения / И.М. Клубков // Вестник ДГТУ. – 2001. – № 1(7). – С. 9. [Электронный ресурс]. – Режим доступа: <http://ocrai.narod.ru/vectory.html> (20.11.2017).
- [7] Fisher, R. Skeletonization / R. Fisher, S. Perkins, A. Walker, E. Wolfart // Medial Axis Transform. – 2003. [Электронный ресурс]. – Режим доступа: <http://homepages.inf.ed.ac.uk/rbf/HIPR2/skeleton.htm> (20.11.2017).
- [8] Моделирование структуры технологического процесса [Электронный ресурс]. – Режим доступа: <https://lektsia.com/4x87a2.html> (20.11.2017).
- [9] Мишенев, В.С. Использование волнового алгоритма для построения скелета растрового изображения / В.С. Мишенев, М.А. Кудрина // Труды Международного симпозиума Надежность и качество. – 2017. – Т. 1. – С. 378-381.
- [10] Регрессия Деминга [Электронный ресурс]. – Режим доступа: [https://ru.wikipedia.org/wiki/Регрессия\\_Деминга](https://ru.wikipedia.org/wiki/Регрессия_Деминга) (20.11.2017)
- [11] Glaister, P. Least squares revisited / P. Glaister // The Mathematical Gazette. – 2001. – № 85. – С. 104-107.



## Wave skeletonization algorithm of raster images

M.A. Kudrina<sup>1</sup>, V.S. Mishenev<sup>1</sup>

<sup>1</sup>Samara National Research University, Moskovskoe Shosse 34A, Samara, Russia, 443086

**Abstract.** This article describes the wave algorithm for binary image skeletonization. In the article we provide the detailed description of the schemes of the main algorithms (algorithm of primary graph creation and algorithm of the wave front division into affiliated waves). For storage of the graph structure we offer to use nested structure that allows save memory and make changing of the graph structure easier. For primary optimization of the graph in the method of consecutive approximations we offer to use orthogonal Deming regression.

**Keywords:** wave algorithm, binary image skeletonization, thinning algorithm, vectorization of raster images.