

# Реализация Пространственных Фильтров с Использованием Аппаратных Ресурсов Видеокарты для Обработки Изображений со Спутника SATUM

Лазэр Д., Капканарь И., Грицков С., Шестакова Т.

Технический Университет Молдовы

Кишинев, Молдова

gritscov@gmail.com

**Abstract** — This paper describes implementation of image filtering based on convolution matrix (mask) for sharpening and blurring with the purpose of noise removal of photographs received from satellite SATUM. To increase significantly image processing speed VGA resources are used. To select most optimal masks for filters, number of iterations, as well as to appreciate effectiveness of the chosen filters an application in Java was elaborated that allows to carry out necessary processing using libraries that allow to involve VGA resources.

**Ключевые слова** — пространственный фильтр, маска свертки, резкость, размытость.

## I. ВВЕДЕНИЕ

Изображения, принятые со спутника, характеризуются значительными искажениями (шум, геометрические искажения), пониженным контрастом и тусклыми цветами. Поэтому при приеме фотографий с молдавского спутника SATUM возникает необходимость повышения их качественных характеристик [1,2]. Сложность этой задачи состоит в необходимости проведения сложных вычислений над большими объемами информации, что задает определенные требования к аппаратным и программным ресурсам.

Для данных целей в спутнике SATUM используются для проведения вычислений программируемые логические схемы – FPGA (Field Programmable Gate Array – программируемая пользователем вентильная матрица), позволяющие сократить стоимость и габариты системы обработки, а также время обработки. Однако предварительно необходимо подобрать методы обработки изображений, оптимальные для данной задачи, которые затем будут реализовываться аппаратно [3,4].

Существующие программные обеспечения позволяют проводить обработку изображений, однако зачастую выбор методов и параметров фильтров ограничен. Для решения данной задачи было разработано приложение, выполняющее обработку изображений на основе методов, оптимальных для обработки фотографий со спутника SATUM и позволяющее выполнять подбор параметров для

используемых фильтров. Выполнение данной обработки на основе центрального процессора компьютера [5] требует значительных временных ресурсов. Рассматриваемое в работе программное обеспечение позволяет сократить в разы скорость вычислений за счет использования аппаратных ресурсов видеокарты. При этом возможность подобрать наиболее оптимальный для нашего случая тип фильтра и его параметры позволит в дальнейшем реализовать выбранные алгоритмы на микросхемах программируемой логики, являющихся основой системы приема данных со спутника SATUM [6].

## II. ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ С ПОМОЩЬЮ МАСОК СВЕРТКИ

Для обработки изображений, переданных со спутника, обычно применяют пространственные или спектральные методы. Для повышения резкости изображений или для удаления шумов с изображений наиболее оптимальными являются пространственные методы. Данные методы по сравнению со спектральными требуют в единицы и десятки раз меньше аппаратных и временных затрат на обработку изображений, обеспечивая при этом приемлемые результаты. Рассмотрим реализацию и применение данных методов обработки изображений.

Основными видами пространственных фильтров, используемых для удаления шумов, являются усредняющие фильтры и фильтры, основанные на порядковых статистиках.

Усредняющие фильтры являются линейными. Они реализуются при помощи операции свертки изображения с ядром (маской) свертки [7]. Данный метод определяет средневзвешенность соседних пикселей. Эти фильтры наиболее эффективны для изображений с равномерным или Гаусовским шумом.

Фильтры, основанные на порядковых статистиках являются нелинейными фильтрами, которые требуют предварительного упорядочения яркостей пикселей, в рассматриваемой окрестности по заданному правилу. Это упорядочивание используется для выбора одного из значений. Фильтры, основанные на порядковых

статистиках (в особенности медианный фильтр) являются наиболее эффективными для изображений с шумом типа salt-and-pepper и шумами Релея.

Рассмотрим более подробно усредняющие фильтры. Данные фильтры изменяют значение каждого пикселя путем вычисления средневзвешенного значения окружающих их пикселей (окна фильтрации). Вес определяется в матричной форме, называемой маской сверки или ядром.

Пусть  $f(x,y)$  – изображение, к которому применяется фильтр и  $w$  – маска свертки (матрица со сторонами  $2a, 2b$ ). Отфильтрованное изображение пикселя  $(x,y)$  задается дискретной сверткой:

$$g(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b w(i, j) \cdot f(x+i, y+j) \cdot \quad (1)$$

где  $x, y$  – координаты текущего пикселя,  $i, j$  – положение пикселя маски.

Маски свертки характеризуются размером группы пикселей. Это называется размером фильтра. Также фильтр характеризуется своей импульсной характеристикой.

Следующая матрица определяет коэффициенты цифрового фильтра размером 3 на 3 пикселя, используемого для повышения резкости изображения:

$-k/8$	$-k/8$	$-k/8$
$-k/8$	$k+1$	$-k/8$
$-k/8$	$-k/8$	$-k/8$

Рис. 1. Матрица для повышения резкости изображения.

На рис. 1 параметр  $k$  определяет степень повышения резкости изображения.

Следующая матрица определяет коэффициенты цифрового фильтра размером 3 на 3 пикселя, используемого для повышения размытости изображения:

$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$
$1/9$	$1/9$	$1/9$

Рис. 2. Матрица для повышения размытости изображения.

Размер матрицы определяет степень размытости. Например, если используется матрица 5 на 5, где каждый элемент равен  $1/25$ , то размытость изображения будет больше, чем при использовании матрицы 3 на 3.

Перейдем к реализации данных фильтров изображений.

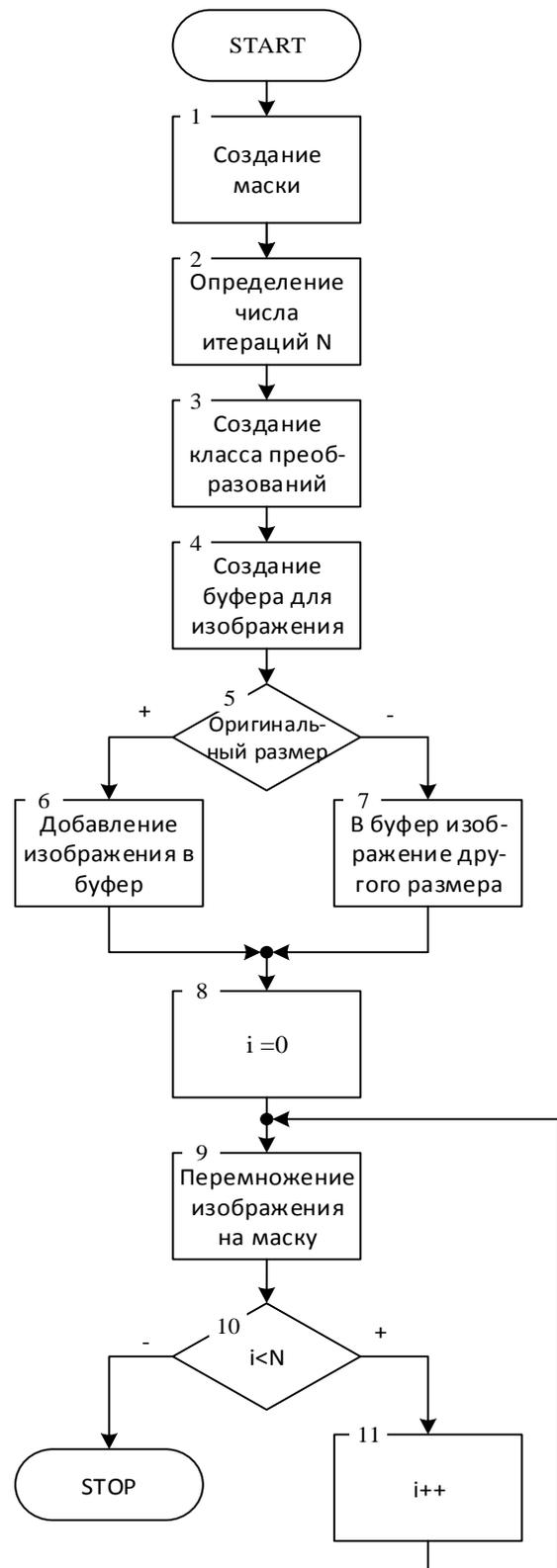


Рис. 3. Алгоритм фильтрации изображения.

### III. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ ФИЛЬТРА С ПОМОЩЬЮ МАСКИ СВЕРТКИ

Для осуществления предложенных методов обработки была разработана программа на языке Java, позволяющая выполнять размытие изображения, повышение резкости и увеличение/уменьшение яркости за счет применения фильтров на основе масок.

Обработка изображения происходит согласно алгоритму, представленному на рис. 3.

Изначально происходит создание массива и запись в него выбранной маски (блок 1), а также задается количество итераций, т.е. сколько раз будет применяться выбранный фильтр (блок 2). После этого создается класс преобразований, в который передается матрица с выбранной маской (блок 3) и буфер временного хранения изображения (блок 4). В зависимости от размера обрабатываемого изображения (блок 5) происходит передача изображения в буфер. Если размеры изображения не совпадают с размером окна, в котором выводится результат, тогда создается буфер по размеру обрабатываемого изображения (блок 7). После этого начинается цикл итераций (блок 9). При этом в каждой итерации происходит перемножение исходной маски и возвращение нового буферизированного изображения.

Ниже представлен код программы, позволяющий выполнить фильтрацию изображения на примере повышения резкости.

```
float [] knet = { -1f, -1f, -1f, //Задание значения маски
                -1f, 9f, -1f,
                -1f, -1f, -1f};

java.awt.image.Kernel ken = new Kernel(3, 3, knet);
// Генерация фильтра 3x3 с маской, указанной выше
java.awt.image.BufferedImageOp conop = new
ConvolveOp(ken);
BufferedImage bufff = null; // Создание буфера для
хранения обрабатываемого изображения на основе
стандартного подкласса BufferedImage
if(original instanceof BufferedImage)
    bufff = (BufferedImage)this.original; // Загрузка в
буфер обрабатываемого изображения
else{
    bufff = new BufferedImage(original.getWidth(null),
original.getHeight(null), BufferedImage.TYPE_INT_RGB);
    bufff.getGraphics().drawImage(this.original, 0, 0,
this);
}
bufff = conop.filter(bufff, null); // применение
фильтра
this.drawing =
bufff.getScaledInstance(drawing.getWidth(null),
drawing.getHeight(null), Image.SCALE_SMOOTH); //
масштабирование изображения перед выводом на экран
this.repaint(); // вывод изображения на экран
```

Результат обработки изображения зависит от применяемой маски и от числа итераций.

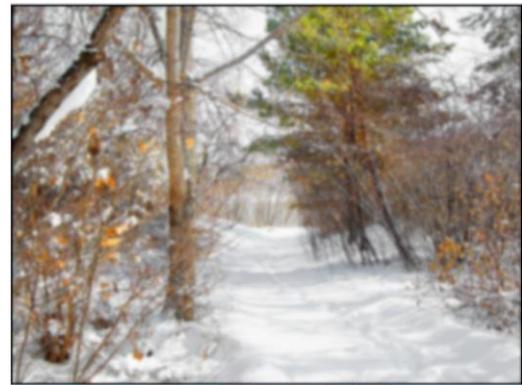
Для устранения шумов и снижения резкости изображения применяется фильтр низких частот (ФНЧ). Задание маски для данного фильтра представлено ниже.

```
float [] knet2 = { 1f/9f, 1f/9f, 1f/9f,
                  1f/9f, 1f/9f, 1f/9f,
                  1f/9f, 1f/9f, 1f/9f};
```

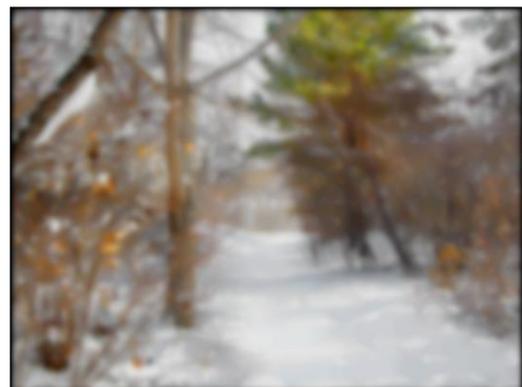
На рис. 4 представлено исходное изображение и результат применения ФНЧ с различным числом итераций.



a)



б)



в)

Рис. 4. Применение ФНЧ к исходному изображению (а) с 10 итерациями (б) и 50 итерациями (в).

Повышение резкости проводится с помощью следующей маски:

```
float [] knet = { -1f, -1f, -1f,
                  -1f, 9f, -1f,
                  -1f, -1f, -1f};
```

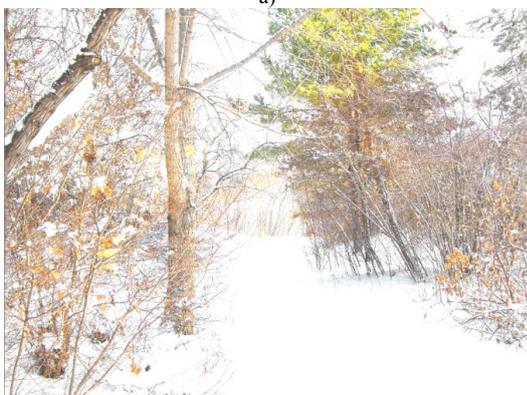
На рис. 5 представлен результат повышения резкости.



Рис. 5. Повышение резкости изображения.



а)



б)

Рис. 6. Изменение значения яркости у исходного изображения: значение яркости -70 (а) и значение яркости 70 (б).

С помощью библиотечных модулей Java можно так же выполнять изменение яркости изображения. Результат представлен на рис. 6.

#### IV. ВЫВОД

Для подбора масок фильтрации для системы обработки изображений, принятых со спутника SATUM, было

разработано приложение на языке Java. Данное приложение использует аппаратные ресурсы видеокарты, что существенно сокращает время обработки изображений по сравнению с временем обработки на основе центрального процессора компьютера. Для улучшения качественных характеристик данных изображений (например, резкость) и устранения шумов были использованы пространственные методы обработки изображений. Реализация приложения на основе данных методов сокращает аппаратные и временные затраты на обработку по сравнению с приложениями на основе частотных методов. Данные особенности разработанного приложения позволят в дальнейшем реализовать разработанный в работе алгоритм на базе FPGA

#### БИБЛИОГРАФИЯ

- [1] С. Грицков, Г. Сорокин, В. Унгуриян, И. Чебан, Г. Бодян, “Дистанционное зондирование Земли студенческим спутником SATUM в режиме реального времени,” Информационные технологии в мире коммуникаций, МТУСИ, Москва, 2013, с. 54-61.
- [2] С. Грицков, И. Чебан, Т. Шестакова, Ю. Костин, “Система фотографирования, обработки и передачи изображений со студенческого нано спутника SATUM,” СИЭТ-2013, Одесса, 2013, с. 266-269.
- [3] И.А. Каляев, И.И. Левин, Е.А. Семерников, В.И. Шмойлов. “Реконфигурируемые мультимедийные вычислительные структуры,” Ростов-на-Дону: ЮНЦ РАН, 2008, 393 с.
- [4] А.Н. Заливин, И.В. Зув. “Использование производительных плис для обработки изображений”, Научные ведомости Белгородского государственного университета, Выпуск № 8-1 (179), том 30, 2014.
- [5] A. Cosolapov, “Prelucrarea imaginilor captate de microsatelitul SATUM cu aplicarea Hard-urilor în baza FPGA,” Lucrarea de master, UTM, FIMET, 2012.
- [6] В. Унгуриян, Г. Сорокин, Г. Бодян, С. Грицков, “Реализация прямой цифровой модуляции/демодуляции BPSK-сигналов,” СИЭТ-2013, Одесса, 2013, с. 258-261.
- [7] А. В. Тамьяров, Р. В. Шестов, “Анализ методов предварительной обработки изображения на основе усредняющих фильтров,” Вестник волжского университета им. В.Н. Татищева, Выпуск № 18, 2011.
- [8] В.Т. Фисенко, Т.Ю. Фисенко, “Компьютерная обработка и распознавание изображений,” учебное пособие, Санкт-Петербургский государственный университет информационных технологий, механики и оптики, 2008, 192 стр.
- [9] Р.А. Шовенгердт, “Дистанционное зондирование. Модели и методы обработки изображений. Часть 1,” М.: Техносфера, 2010. - 560 с. - ISBN: 978-5-94836-244-1.
- [10] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, “Digital Image Processing Using MATLAB, 2nd edition,” Gatesmark Publishing, 2009, 827 p. - ISBN: 9780982085400.