

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное  
образовательное учреждение высшего образования  
«Комсомольский–на–Амуре государственный университет»

На правах рукописи



Лозинский Василий Сергеевич

**Разработка программного обеспечения для восстановления смазанных  
или размытых зашумленных изображений**

Направление подготовки

01.04.02 «Прикладная математика и информатика»

АВТОРЕФЕРАТ  
МАГИСТЕРСКОЙ ДИССЕРТАЦИИ

2019

Работа выполнена в ФГБОУ ВО «Комсомольский-на-Амуре  
государственный университет»

Научный руководитель: кандидат физико-математических наук, доцент, доцент кафедры «Прикладная математика и информатика»  
Егорова Юлия Георгиевна

Рецензент: кандидат технических наук, доцент кафедры математики факультета информационных технологий, математики и физики ФГБОУ ВО «Амурский гуманитарно-педагогический государственный университет»  
Севастьянов Антон Мамиевич

Защита состоится 25 июня 2019 года в 9 часов 50 мин на заседании государственной экзаменационной комиссии по направлению подготовки 01.04.02 «Прикладная математика и информатика» в Комсомольском-на-Амуре государственном университете по адресу: 681013, г. Комсомольск-на-Амуре, пр. Ленина, 27, ауд. 312/3.

Автореферат разослан 17 июня 2019 г.

Секретарь ГЭК



Ю.Г.Егорова

## ОБЩАЯ ХАРАКТЕРИСТИКА ДИССЕРТАЦИОННОЙ РАБОТЫ

*Актуальность темы.* Исследования в этой области имеют множество вариантов применения на практике в таких сферах как: любительская или бытовая фотография, применение камер наблюдения, использование фотографий или видеозаписей при ведении расследований или в качестве доказательств в суде, космическая фотосъемка.

Достигнутые на данный момент практические результаты ещё далеки от идеальных. К повышению качества фотографий существуют различные подходы, которые могут принадлежать к совершенно разным отраслям науки. В работе рассматривается один из методов прямого восстановления.

*Целью* исследования:

- Исследование существующих возможностей по восстановлению размытых и зашумленных изображений;
- Реализация алгоритма восстановления размытого изображения на основе преобразования Фурье.

Для достижения указанной цели поставлены следующие *задачи*:

- Изучить существующие возможности по восстановлению размытых и зашумленных изображений;
- Определиться со средой написания программного комплекса;
- Разработать интерфейс программного комплекса;
- Реализовать алгоритм восстановления размытого изображения;
- Проанализировать полученные результаты и сделать выводы об эффективности алгоритма

*Объектом* исследования смазанное или зашумленное изображение.

*Предметом* исследования является программный комплекс позволяющая восстанавливать смазанные и зашумленные изображения.

Для решения поставленных задач использовались *следующие* методы исследования: теоретические (анализ) и эмпирические (изучение литературы и результатов деятельности).

*Научная новизна* исследования заключается в следующем:

Создан программный комплекс по восстановлению зашумленных и размытых изображений

*Достоверность и обоснованность результатов исследования.*

Программа должна быстро обработать отдельные участки изображения и восстановить целостную структуру картинки, ориентируясь при этом на тип дефекта (расфокусировка или размытие Гаусса).

Разработанное программное обеспечение тестировалось как на реальных изображениях и предоставляло оптимальные результаты.

*Практическая значимость* заключается в восстановлении зашумленных и смазанных изображений.

*Апробация результатов.*

Результаты работы докладывалась на:

«Лозинский, В.С. Разработка программного обеспечения для восстановления смазанных или размытых зашумленных изображений / В.С. Лозинский, Ю.Г. Егорова // Сборник материалов конференции : 2-ая всерос. национальная науч. конф. студентов, аспирантов и молодых учёных «Молодёжь и наука: актуальные проблемы фундаментальных и прикладных исследований», Комсомольск-на-Амуре, 8-12 апр. 2019 г.

*Публикации.* По результатам выполненных в диссертации исследований автором опубликовано 1 работы.

Структура и объем. Магистерская диссертация состоит из введения, двух глав, заключения и списка литературы. Объем работы –46 страницы, в том числе 18 рисунков и 2 приложения.

## ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

*Введение* раскрывает актуальность темы, определяются цели и задачи исследования, объект, предмет, указывается научная новизна.

В *первой главе* описывается алгоритм.

Модель процесса искажения

Рассмотрим только полутоновые черно-белые изображения в предположении, что для обработки полноцветного изображения достаточно повторить все необходимые шаги для каждого из цветовых каналов RGB.

Введем следующие обозначения:

$f(x, y)$  – исходное неискаженное изображение

$h(x, y)$  – искажающая функция

$n(x, y)$  – аддитивный шум

$g(x, y)$  – результат искажения, то есть то, что мы наблюдаем в результате (смазанное или расфокусированное изображение)

Сформулируем модель процесса искажения следующим образом:

$$g(x, y) = h(x, y) * f(x, y) + n(x, y) \quad (1)$$

Задача восстановления искаженного изображения заключается в нахождении наилучшего приближения  $f(x, y)$  исходного изображения. Рассмотрим каждую составляющую более подробно. С  $f(x, y)$  и  $g(x, y)$  все достаточно понятно. А вот про функцию  $h(x, y)$  нужно рассказать подробнее. В процессе искажения каждый пиксель исходного изображения превращается в пятно для случая расфокусировки и в отрезок для случая простого смаза. Либо же можно сказать наоборот, что каждый пиксель искаженного изображения «собирается» из пикселей некоторой окрестности исходного изображения. Все это друг на друга накладывается, и в результате мы получаем искаженное изображение. То, по какому закону размазывается или собирается один пиксель, и называется функцией искажения. Другие синонимы – PSF (Point spread function, т.е. функция распределения точки), ядро искажающего оператора, kernel и другие. Размерность этой функции,

как правило, меньше размерности самого изображения – к примеру, в начальном рассмотрении примера размерность функции была 2, так как каждый пиксель складывался из двух.

### *Модель процесса искажения*

Рассмотрим, как выглядят искажающие функции. Здесь и далее будем использовать ставший уже стандартным для таких целей инструмент – Matlab, он содержит в себе все необходимое для самых разнообразных экспериментов с обработкой изображений (и не только) и позволяет сосредоточиться на самих алгоритмах, перекладывая всю рутинную работу на библиотеки функций. Впрочем, за это приходится расплачиваться производительностью.

Операция применения искажающей функции к другой функции (к изображению, в данном случае) называется сверткой (convolution), то есть некоторая область исходного изображения сворачивается в один пиксель искаженного изображения. Обозначается через оператор «\*». Математически для изображения  $f$  с размерами  $M \times N$  и искажающей функции  $h$  с размерами  $m \times n$  это записывается так:

$$g(x, y) = h(x, y) * f(x, y) = \sum_{i=-a}^a \sum_{j=-b}^b h(i, j) f(x + i, y + j) \quad (2)$$

где  $a = (m - 1) / 2$ ,  $b = (n - 1) / 2$ . Операция, обратная свертке, называется деконволюцией (deconvolution), и решение такой задачи весьма нетривиально.

### *Модель шума*

Осталось рассмотреть последнее слагаемое, отвечающее за шум,  $n(x, y)$  в формуле (1). Причины шума в цифровых сенсорах могут быть самыми разными, но основные это – тепловые колебания и темновые токи. На величину шума также влияет ряд факторов, таких как значение ISO, тип матрицы, размер пикселя, температура, электромагнитные наводки и пр. В большинстве случаев шум является Гауссовым (который задается двумя

параметрами – средним и дисперсией), а также является аддитивным, не коррелирует с изображением и не зависит от координат пикселя. Последние три предположения являются очень важными для дальнейшей работы.

### *Теорема о свёртке*

Вернемся теперь к первоначальной постановке задачи восстановления – нам необходимо обратить свертку, при этом не забывая про шум. Из формулы (2) видно, что получить  $f(x, y)$  из  $g(x, y)$  не так-то просто – если решать задачу напрямую, то получится огромная система уравнений. Но на помощь к нам приходит преобразование Фурье, не будем подробно на нем останавливаться, по этой теме уже было сказано немало. Так вот, есть такая теорема о свертке, которая гласит, что операция свертки в пространственной области эквивалентна обычному умножению в частотной области (причем умножение поэлементное, а не матричное). Соответственно, операция обратная свертке эквивалентна делению в частотной области, то есть это можно записать как:

$$h(x, y) * f(x, y) \Leftrightarrow H(u, v)F(u, v) \quad (3)$$

где  $H(u, v)$ ,  $F(u, v)$  – Фурье-образы соответствующих функций. Значит, процесс искажения из формулы (1) можно переписать в частотной области как:

$$G(u, v) = H(u, v)F(u, v) + N(u, v) \quad (4)$$

Во *второй главе* реализация алгоритма.

Сделали с помощью фотоаппарата смазанное фото (рисунок 3).



Рисунок 1 – Смазанное изображение

Процесс смаза можно отобразить в виде выражения:

$$\mathbf{B} = \mathbf{K} * \mathbf{L} + \mathbf{N}$$

где  $\mathbf{B}$  — наблюдаемое смазанное изображение,  $\mathbf{K}$  — ядро размытия (траектория смаза), kernel  $\mathbf{L}$  — исходное неразмытое изображение (скрытое)  $\mathbf{N}$  — аддитивный шум, символ  $*$  - операция свёртки (convolution).

Чтобы получить конкретное решение, нужно вводить ограничения, как-то описывать модель того, к чему мы стремимся.

То есть нужно больше информации. Одним из вариантов этого является задание целевой функции — это такая функция, значение которой тем выше (в простом случае), чем ближе получаемый результат к желаемому.

Исследования показали, что, несмотря на то, что реальные изображения имеют большой разброс значений отдельных пикселей, градиенты этих значений имеют вид распределения с медленно убывающими границами (Heavy-tailed distribution). Такое распределение имеет пик в окрестности нуля и, в отличие от гауссового распределения, имеет значительно большие вероятности больших значений.

Это совпадает с интуитивным представлением, что на реальных изображениях в большинстве случаев присутствуют большие области более-менее постоянной яркости, которые заканчиваются объектами с резкими и средними перепадами яркости (рисунки 2-3).



Рисунок 2 – Гистограмма градиентов для резкого изображения

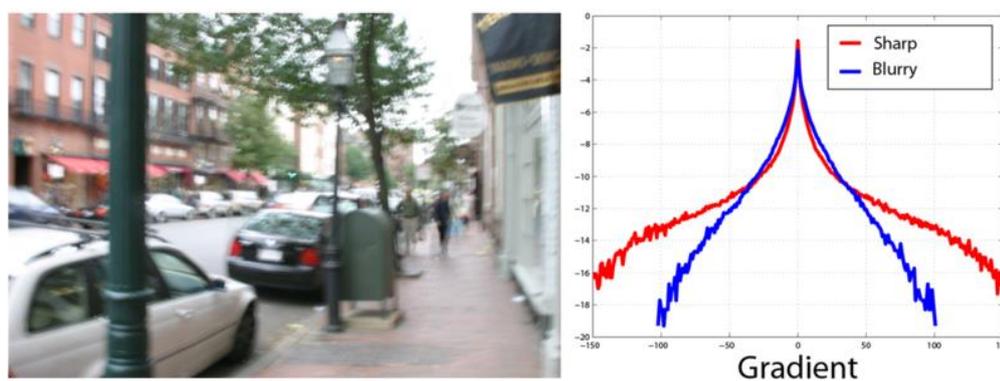


Рисунок 3 – Гистограмма градиентов для размытого изображения

Таким образом, мы получили инструмент, которые позволяет нам измерить «качество» получаемого результата с точки зрения четкости и похожести на реальное изображение.

Теперь мы можем сформулировать основные пункты для построения завершённой модели:

- 1 Ограничения, накладываемые моделью искажения  $V = K * L + N$
- 2 Целевая функция для результата реконструкции — насколько похожа гистограмма градиентов на теоретическое распределение с медленно убывающими границами. Обозначим ее как  $p(LP)$
- 3 Целевая функция для ядра искажения — положительные значения и низкая доля белых пикселей (т.к. траектория смаза обычно представлена тонкой линией). Обозначим ее как  $p(K)$ .

Объединим все это в общую целевую функцию следующим образом:

$$\langle \log \frac{q(\nabla \mathbf{L}_p)}{p(\nabla \mathbf{L}_p)} \rangle_{q(\nabla \mathbf{L}_p)} + \langle \log \frac{q(\mathbf{K})}{p(\mathbf{K})} \rangle_{q(\mathbf{K})} + \langle \log \frac{q(\sigma^{-2})}{p(\sigma^2)} \rangle_{q(\sigma^{-2})}$$

где  $q(\mathbf{L}_p)$ ,  $q(\mathbf{K})$  – распределения, получаемые подходом.

### *Пирамидальный подход*

Решением этой проблемы является итерационный подход к построению ядра искажения.

Вначале из входного размытого изображения мы строим пирамиду изображений с разным разрешением. От самого маленького размера до исходного размера.

Далее мы инициализируем алгоритм с помощью ядра размером  $3 \times 3$  с одним из простых шаблонов — вертикальная линия, горизонтальная линия или гауссово пятно. В качестве универсального варианта можно выбрать последний — гауссово пятно.

Используя алгоритм оптимизации, описанный выше, мы улучшаем оценку ядра, используя самый маленький размер изображения в построенной пирамиде.

После этого мы ресайзим полученное уточненное ядро до, скажем,  $5 \times 5$  пикселей и повторяем процесс уже с изображением следующего размера.

Таким образом, на каждом шаге мы чуть-чуть улучшаем ядро и в результате получаем весьма точную траекторию смаза.

Продемонстрируем итеративное построение ядра на примере (рисунки 4-5).



Рисунок 4 – Исходное изображение для пирамидального подхода

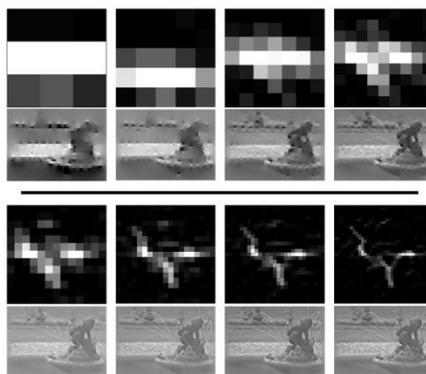


Рисунок 5 – Процесс уточнения ядра

Первая и третья строки показывают оценку ядра на каждом уровне пирамиды изображений. Вторая и четвертая строки — оценка неискаженного изображения.



Рисунок 6 – Результат

## СПИСОК ПУБЛИКАЦИЙ ПО ТЕМЕ ИССЛЕДОВАНИЯ

«Лозинский, В.С. Разработка программного обеспечения для восстановления смазанных или размытых зашумленных изображений / В.С. Лозинский, Ю.Г. Егорова // Научно-техническое творчество аспирантов и студентов: материалы 47-й науч.- техн. конф. студентов и аспирантов. - Комсомольск-на-Амуре : ФГБОУ ВО «КНАГТУ», 2019. - С. 209-211.