

УДК 004.056

Корниенко Н. С., Цололо С.А., Юсупова К. Б.

Донецкий национальный технический университет, г.Донецк
кафедра компьютерной инженерии

РЕКОНСТРУКЦИЯ СМАЗАННЫХ И РАСФОКУСИРОВАННЫХ ИЗОБРАЖЕНИЙ

Аннотация

Корниенко Н. С., Цололо С.А., Юсупова К.Б. Реконструкция смазанных и расфокусированных изображений. Выполнен анализ особенностей деконволюции Винера для восстановления смазанных и расфокусированных изображений. Предложена программная реализация фильтра Винера. Приведен способ оптимизации работы фильтра в программном продукте, а также результаты работы разработанного приложения.

Ключевые слова: *методы предобработки изображений, выделение границ объектов, оконтуривание изображений.*

Введение

На сегодняшний день существуют множество методов или фильтров, которые призваны улучшить фотоизображения или же сделать их красивее и художественнее. Но бывают ситуации, когда речь идет о прямой необходимости улучшить изображение с технической точки зрения, например, – для возможности просто прочесть смазанный текст. В этом случае остается лишь два варианта. Первый и очевидный – повторить фотосъемку и получить качественную фотографию необходимого объекта. Иногда это возможно, но, как правило, – такой возможности нет. Второй вариант – использовать программную обработку, а именно фильтры, специализацией которых является восстановление изображения после тех или иных искажений. Таким образом, в работе авторами предлагается реализация одного из фильтров, способного восстановить искаженное (смазанное) изображение.

Целью работы является разработка способа реконструкции смазанных и расфокусированных изображений, в основе которого лежит использование фильтра Винера. В работе предлагается программная реализация фильтра, а также рассмотрены основные пути повышения быстродействия программной реализации.

Анализ задачи

Любое искаженное изображение содержит информацию об оригинале. По сути, будь то расфокусировка, размытие или же смаз, – информация о конкретном пикселе не исчезает, а лишь изменяется по определенному закону. Для определения оригинальной информации необходимо сделать деконволюцию (обратную свертку).

Для деконволюции в данной работе будет использоваться один из основных методов – фильтр Винера. При это необходимо иметь учитывать, что вид ядра деконволюции определяется типом искажения.

Фильтр Винера

Фильтр Винера представляет изображение и шум как случайные процессы и находит такую оценку для неискаженного изображения, чтобы среднеквадратическое отклонение этих величин было минимальным. Минимум этого отклонения достигается на функции в частотной области:

$$\hat{F}(uv) = \left(\frac{1}{H(u, v) \left(|H(u, v)|^2 + S_{\eta}(u, v) / S_f(u, v) \right)} \right) G(u, v) \quad (1)$$

Преобразование формулы (1) приводит к следующему результату:

$$\hat{F}(uv) = \left(\frac{|H(u, v)|^*}{|H(u, v)|^2 + S_{\eta}(u, v) / S_f(u, v)} \right) G(u, v), \quad (2)$$

где:

\hat{F} – Фурье-образ для искомого изображения;

H^* – комплексное сопряженное Фурье-образа ядра;

H^2 – квадрат модуля Фурье-образа ядра;

S_{η} / S_f – энергетические спектры шума и исходного изображения;

G – Фурье-образ исходного изображения.

С учетом того, что соотношение спектра шума и исходного изображения однозначно точно определить невозможно, в формуле (2) примем S_{η} / S_f за некую константу k , которую можно приблизительно охарактеризовать как соотношение «сигнал-шум»:

$$\hat{F}(uv) = \left(\frac{|H(u, v)|^*}{|H(u, v)|^2 + k} \right) G(u, v). \quad (3)$$

Выбор функции рассеяния точки

Реализация формулы в программировании не является чем-то сложным, однако следует внести некоторые разъяснения. В предлагаемой работе все преобразования Фурье выполняются при помощи свободной библиотеки FFTW3 [2]. При этом ядром является функция распределения точки искаженного изображения (Point spread function — PSF) [3].

На практике же реализовать программный продукт, который будет однозначно определять PSF по полученному искаженному изображению, весьма непросто, в некоторых случаях – невозможно. Поэтому в данном программном продукте в качестве ядра (PSF) для восстановления расфокусированного изображения используется окружность на черном фоне,

закрашенную белым, а для смазанных изображений – отрезок белого цвета на черном фоне, длина и радиус поворота которого зависит от направления и радиуса смаза. Все параметры ядра регулируется вручную через программный интерфейс.

Пример PSF для расфокусированного и смазанного изображения приведены на рисунках 1а и 1б соответственно.

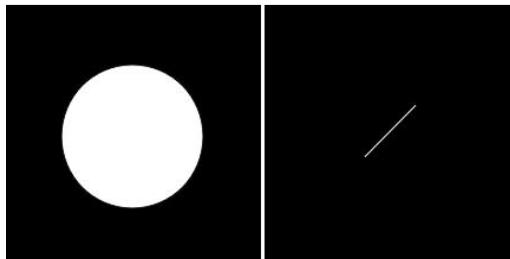


Рисунок 1 – а) функция распределения точки для расфокусированного изображения, б) функция распределения точки для смазанного изображения.

Реализация фильтра

Программа реализации написана при помощи кросс-платформенного инструмента разработки программного обеспечения Qt [5] на языке C++. При разработке были использованы библиотеки как стандартные библиотеки Qt, так и специализированная библиотека FFTW – для вычислений Фурье преобразований. Программа реализована в объектно-ориентированном стиле, что в дальнейшем поспособствует расширению программы. В программе использованы следующие основные классы:

1. Class MainWindow – класс для создания и управления интерфейсом пользователя.
2. Class Filter – абстрактный класс, содержащий общие методы всех фильтров.
3. Class Wiener – класс, наследуемый от класса Filter, обеспечивающий непосредственную работу фильтра Винера.

Для реализации фильтра Винера необходимо реализовать:

- генератор PSF;
- методы для получения прямого и обратного преобразования Фурье изображения;
- реализовать непосредственное решение формулы фильтра Винера.

Сгенерировать PSF (в нашем случае – идеальное PSF) не составляет труда. Достаточно лишь с помощью языка программирования или же вручную в редакторе нарисовать окружность (линию) с требуемым радиусом (требуемой длины и углом поворота, рис. 1).

Основное окно разработанной программы приведено на рисунке 2.

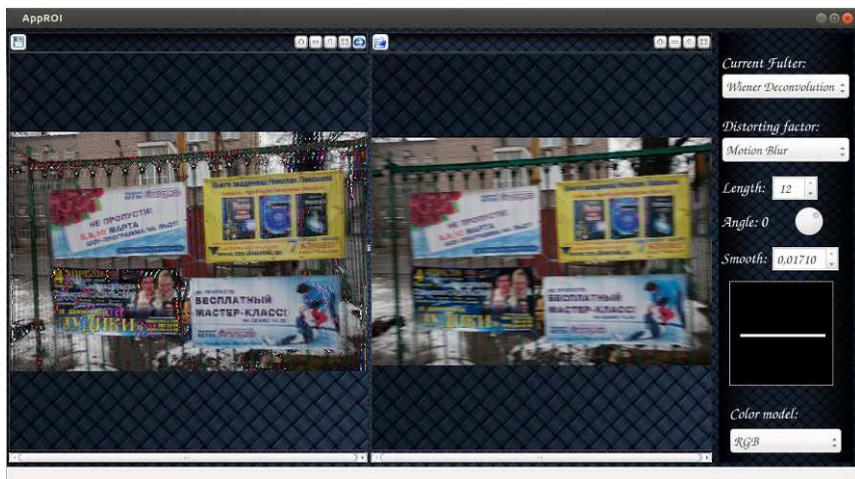


Рисунок 2 – Основное окно программы

Оптимизация быстродействия программы

Поскольку выполнение алгоритма деконволюции для изображения 1200×800 без оптимизации составляет 200мс, то возникла необходимость оптимизировать программный код по времени.

Стоит отметить, что параметр «unsigned flags» для функции «fftw_plan_dft_2d» имеет множество вариантов, однако мы рассмотрим два основных: FFTW_ESTIMATE и FFTW_MEASURE. Различие данных флагов заключается в том, что

- при FFTW_ESTIMATE преобразование Фурье выполняется каждый раз снова и требует приблизительно одинаковое время;

- при использовании FFTW_MEASURE выполняется оптимальный расчет, однако при первом запуске время выполнения преобразования Фурье возрастает вплоть до одной секунды, однако все последующие преобразования требуют на порядок меньше времени, чем аналогичные расчеты при использовании флага FFTW_ESTIMATE.

На рисунке 3 приведена диаграмма зависимости скорости фурье преобразования для изображения 1200×800 от используемого флага FFTW_MEASURE или FFTW_ESTIMATE.

Так же сложно не заметить, что в формуле используется преобразование Фурье искаженного изображения, которое в зависимости от настроек не изменяется. Это дает возможность избавиться от вычисления образа Фурье входного изображения на этапе работы фильтра. Достаточно лишь при открытии изображения выполнить данное преобразование, а далее использовать полученный результат.

В итоге получается, что при использовании FFTW_ESTIMATE для полной работы фильтра требуется порядка 130 мс, а при FFTW_MEASURE для первого расчета 1500мс, для всех последующих – 70мс.

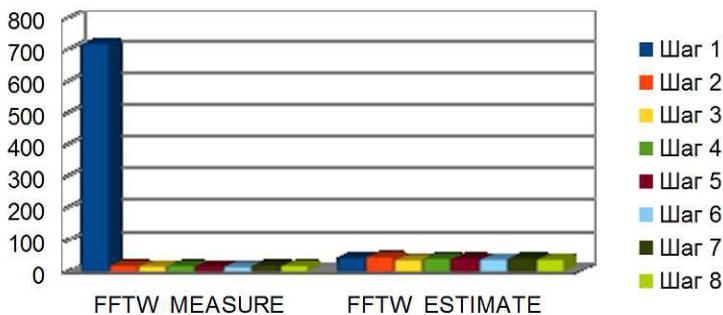
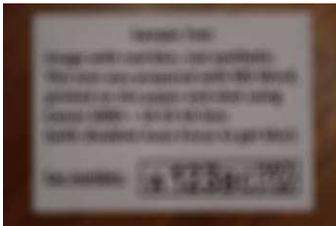
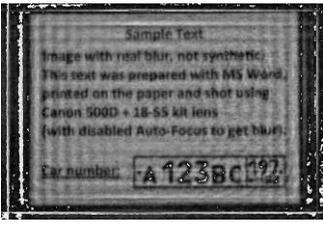
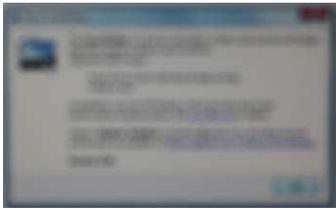


Рисунок 3 – Время выполнения преобразования Фурье, в зависимости от используемого флага.

Тестовые результаты программы

В таблице 1 приведены тестовые результаты работы программы. Приведены исходное и результирующее изображения, а также использованные настройки для каждого из изображений.

Таблица 1– Результаты обработки изображений

Исходное изображение	Восстановленное изображение	Настройки фильтра
		K = 0.0017 Радиус = 28px
		K = 0.0010, Радиус = 13px

Продолжение таблицы 1



Проанализировав таблицу, можно сделать вывод, что реализованный метод, конечно же, не позволяет полностью восстановить исходное смазанное изображение. Однако для эффективного применения этого и не требуется, достаточно того, что после обработки появилось возможным идентифицировать отдельные фрагменты рисунков (текст, фигуры). Также стоит отметить, что значение константы K (соотношение «сигнал-шум») в общем случае не превышает значения «0.01».

Выводы

Работа посвящена актуальной задаче реконструкции изображений. В результате было разработано программное обеспечение, способное корректировать смазанные и расфокусированные изображения. рамках предложенной тематики в дальнейшем планируется как разработка как других фильтров, так и методов, способных вычислять оптимальное PSF, а также дальнейшая оптимизация разработанной программы по быстрдействию.

Список литературы

1. Винер деконволюция. [электронный ресурс]. – Режим доступа: http://en.wikipedia.org/wiki/Wiener_deconvolution
2. Свободная библиотека для БПФ [электронный ресурс]. – Режим доступа: <http://www.fftw.org/>
3. Статья о восстановлении смазанных и расфокусированных изображений [электронный ресурс]. – Режим доступа: <http://habrahabr.ru/post/136853/>