

02, февраль 2016

УДК 004.352.242

Оптимизация решения задачи восстановления изображения методом Тихонова

Данилина Е.А., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы автоматизированного проектирования»*

Елфимов В.Т., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы автоматизированного проектирования»*

Научный руководитель: Волосатова Т.М., к.т.н., доцент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана
кафедра «Системы автоматизированного проектирования»*

bauman@bmstu.ru

1. Постановка задачи

Восстановление искаженных изображений является одной из наиболее интересных и важных проблем в задачах обработки изображений – как с теоретической, так и с практической точек зрения. При съемке быстродвижущихся объектов или съемке в движении происходит смазывание или расфокусировка изображения.

Процесс искажения изображения в общем виде можно описать следующей формулой:

$$g(x, y) = H[f(x, y)] + \eta(x, y),$$

где $g(x, y)$ - искаженное изображение (физический сигнал на выходе оптической системы);
 $f(x, y)$ - идеальное (неискаженное) аналоговое изображение; H - искажающий оператор оптической системы, $\eta(x, y)$ - аддитивный шум.

Функции $g(x, y)$, $f(x, y)$, $\eta(x, y)$ являются непрерывными функциями от непрерывных пространственных координат, поскольку преобразование к дискретным величинам произойдет только после прохождения процедуры аналого-цифрового преобразования сигнала.

Имея функцию $g(x,y)$, обладая информацией об искажающем операторе H оптической системы и зная основные характеристики аддитивного шума $\eta(x,y)$, можно построить некоторое оптимальное приближение $f(x,y)$. При этом чем больше имеется информации об операторе H и аддитивном шуме $\eta(x,y)$, тем точнее может быть построено приближение $f(x,y)$. Схема, моделирующая процесс искажения и коррекцию изображения, представлена на Рис. 1.

Если выполняются условия

- линейности

$$H(af_1(x,y) + bf_2(x,y)) = aH(f_1(x,y)) + bH(f_2(x,y));$$

- трансляционной инвариантности (изопланатизма)

$$H(f(x-\alpha, y-\beta)) = g(x-\alpha, y-\beta),$$

можно показать математически, что искаженное изображение представимо в пространственной области в следующем виде:

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

где $h(x,y)$ - импульсная (искажающая) функция (пространственное представление искажающего оператора H).

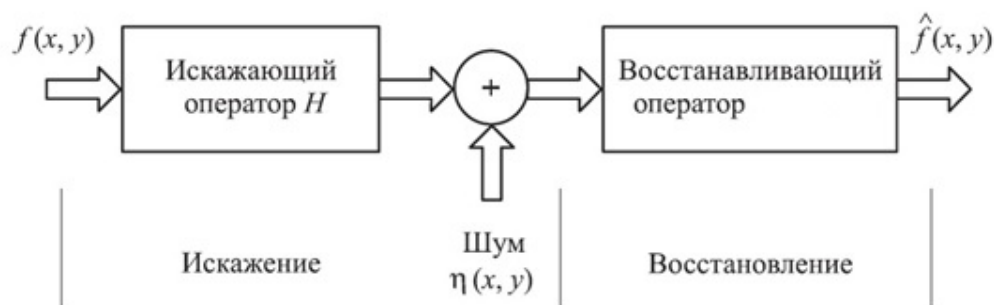


Рис. 1. Моделирование процесса искажения и коррекции изображения

В различных источниках функцию $h(x,y)$ называют также функцией пространственного отклика, ядром искажающего оператора, функцией *kernel*, функцией рассеяния точки (*PSF*, от англ. *Point Spread Function*).

Операция применения искажающей функции к другой функции (в данном случае к изображению $f(x,y)$) называется сверткой (от англ. *convolution*). В формуле (1) эта операция обозначена символом $*$. Смысл операции свертки состоит в том, что в

формировании значения точки A с координатами (x_0, y_0) , принадлежащей искаженному изображению g , принимают участие точки из некоторой окрестности δ точки B с координатами (x_0, y_0) , принадлежащие исходному изображению f , причем размер окрестности δ и вклад точек из окрестности B в формирование значения точки A целиком определяется искажающей функцией.

Для пояснения вышесказанного рассмотрим пример. Предположим, что ведется работа с растровым (пиксельным) изображением, имеющим разрешение $M \times N$ – это размер линейного поля зрения. Разрешение определяется эффективным размером импульсного отклика. В таком случае переменные x, y будут определять порядковый номер пикселя в изображении, следовательно, $x \in 0, 1, \dots, M$; $y \in 0, 1, \dots, N$. Пусть окрестность δ имеет размер $m \times n$. Тогда математически процесс применения операции свертки к изображению записывается следующим образом:

Линейная свертка – основная операция цифровой обработки сигналов, особенно в режиме реального времени. Для двух конечных причинных последовательностей $h(n)$ и $y(k)$ длиной соответственно N и K свертка определяется выражением:

$$s(k) = h(n) * y(k) = \sum_{n=0}^N h(n)y(k-n),$$

где $*$ - символные обозначения операции свертки.

Как правило, в системах обработки одна из последовательностей $y(k)$ представляет собой обрабатываемые данные (сигнал на входе системы), вторая $h(n)$ – оператор (импульсный отклик) системы, а функция $s(k)$ – выходной сигнал системы. В компьютерных системах с памятью для входных данных оператор $h(n)$ может быть двусторонним: от $-N_1$ до $+N_2$ (например, симметричным: $h(-n) = h(n)$) с соответствующим изменением пределов суммирования, что позволяет получать выходные данные без сдвига фазы частотных гармоник относительно входных данных. При строго корректной свертке с обработкой всех отсчетов входных данных размер:

$$g(x, y) = h(x, y) * f(x, y) = \frac{1}{mn} \sum_{i=-a}^a \sum_{j=-b}^b h(i, j) f(x+i, y+j),$$

где $a = \left\lfloor \frac{m-1}{2} \right\rfloor, b = \left\lfloor \frac{n-1}{2} \right\rfloor$.

На практике часто $a = b$. В этом случае число a называется размерностью искажающей функции. Размерность искажающей функции, как правило, меньше размерности самого изображения.

На Рис. 2 представлен пример искажающей функции для моделирования смазывания изображения (размерность искажающей функции равна 30).

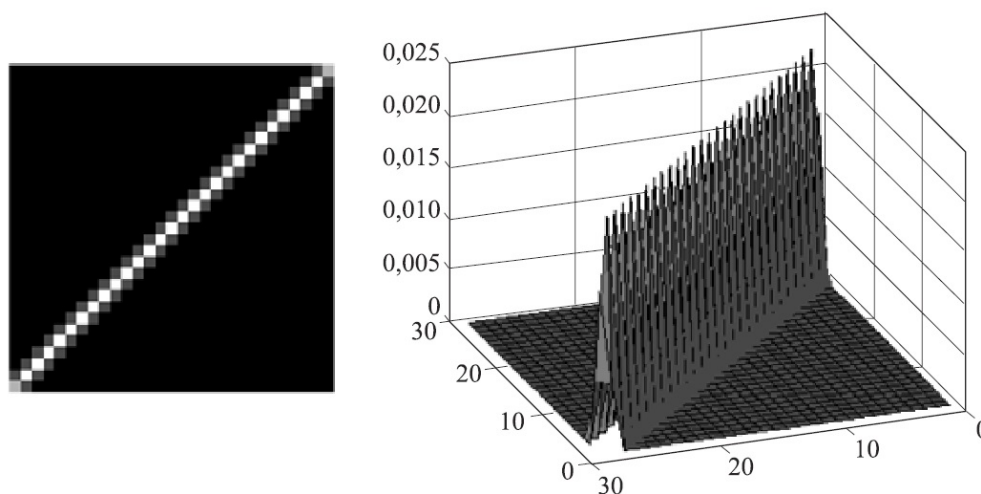


Рис. 2. Искажающая функция для моделирования смазывания изображения за счет бега подстилающей поверхности

На рис. 3 представлен пример искажающей функции для моделирования оптического размытия изображения (размерность искажающей функции равна 30).

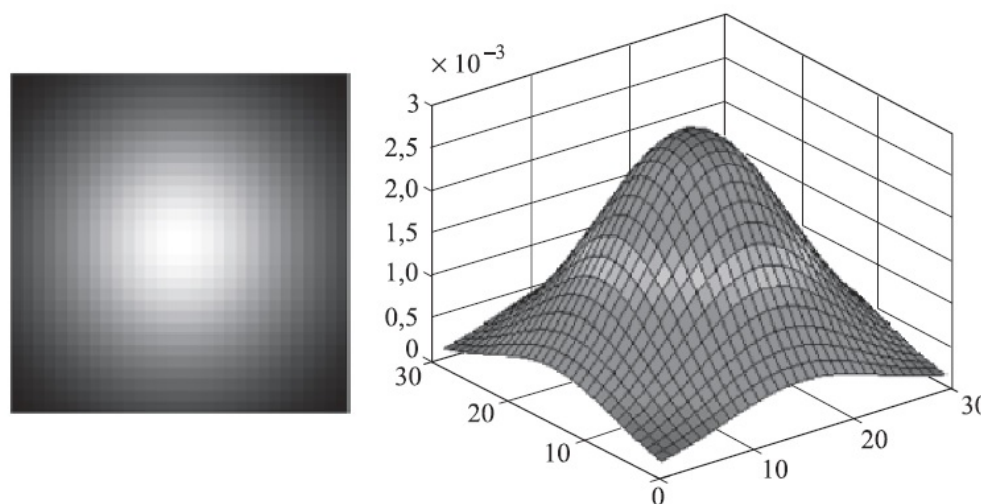


Рис. 3. Искажающая функция для моделирования смазывания размытия по закону Гаусса

В соответствии с теоремой о свертке, в пространственной области свертка функций эквивалентна умножению в частотной области Фурье - преобразований этих функций, поэтому приведенное выше уравнение модели искажения в пространственной области (2) можно записать в эквивалентном представлении в частотной области:

$$G(u, v) = H(u, v)F(u, v) + N(u, v), \quad (3)$$

где заглавными буквами даны соответствующие Фурье-преобразования функций, обозначенных строчными буквами в уравнении свертки в пространственной области (2).

Функцию $H(u, v)$ часто называют оптической передаточной функцией (*OTF*, от англ. *Optical Transfer Function*). Этот термин заимствован из анализа Фурье оптических систем. Очевидно, что функции $h(u, v)$ и $H(u, v)$ переходят одна в другую под действием прямого и обратного Фурье-преобразования.

2. Моделирование искажающей функции

Получение точной оценки искажающей функции оптико-электронной системы — один из важных шагов на пути к восстановлению искаженного изображения.

Существует три основных способа оценки искажающей функции оптико-электронной системы:

- визуальный анализ;
- экспериментальный анализ;
- математическое моделирование.

3. Фильтрация методом Тихонова.

Реализация метода минимизации сглаживающего функционала со связью требует только знания среднего значения и дисперсии шума. Это является важным преимуществом метода, поскольку обычно можно оценить указанные величины на основе заданного искаженного изображения.

Пользуясь определением свертки (2), можно записать выражение (1) в векторном виде:

$$g = Hf + \eta. \quad (10)$$

Пусть изображение $g(x, y)$ имеет размеры $M \times N$. Тогда вектор g формируется таким образом, что первые M его элементов в точности совпадают с первой строкой изображения $g(x, y)$, вторые M элементов — со второй строкой и т. д. Таким образом, в данном случае вектор g будет иметь длину MN . Такую же длину будут иметь векторы f и η . При этом матрица H будет иметь размер $MN \times MN$. Ее элементы определяются значениями h в свертке (1).

Формулировка задачи восстановления в матричном виде облегчает построение методов восстановления. Главной проблемой является чувствительность матрицы к шуму. Один из

способов решения этой проблемы состоит в регуляризации задачи, которая заключается в замене исходной задачи на задачу нахождения экстремума некоторого сглаживающего функционала (данный подход также называется регуляризацией Тихонова). В качестве такого функционала можно использовать квадрат нормы лапласиана

$$C(f) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (\nabla f(x, y))^2$$

с дополнительным ограничением (связью) вида

$$\|g - Hf\|^2 = \|\eta\|^2. \quad (12)$$

Решение оптимизационной задачи (11) в частотной области может быть представлено следующим выражением:

$$\hat{F}(u, v) = \left(\frac{H^*(u, v)}{|H(u, v)|^2 + \gamma |P(u, v)|^2} \right) G(u, v), \quad (13)$$

где параметр регуляризации γ должен быть выбран в соответствии с (12), а функция

$P(u, v)$ есть Фурье-преобразование функции

$$p(x, y) = \begin{vmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{vmatrix},$$

т. е. функции, с помощью которой определяется оператор Лапласа. Подчеркнем, что при обращении параметра регуляризации γ в ноль выражение (13) сводится к инверсной фильтрации. Значения параметра регуляризации γ можно перебирать в интерактивном режиме до тех пор, пока не будет получен приемлемый результат. Однако для получения оптимального решения значение γ должно быть выбрано таким образом, чтобы выполнялось условие связи (12).

Ниже приведена итеративная процедура такого выбора.

Определим вектор невязки r следующим образом:

$$r = g - H\hat{f}. \quad (14)$$

Поскольку решение $\hat{F}(u, v)$ (и соответствующий ему вектор \hat{f}), определяемое по формуле (13), есть функция от параметра γ , вектор невязки r также зависит от этого параметра. Можно показать, что функционал невязки

$$\phi(\gamma) = r^T r = \|r\|^2$$

является монотонной функцией параметра γ . Нам необходимо выбрать параметр γ таким образом, чтобы выполнялось условие:

$$\|\eta\|^2 - a \leq \|r\|^2 \leq \|\eta\|^2 + a, \quad (15)$$

где коэффициент a задает приемлемую точность выполнения результата связи. Если $a = 0$, имеет место $\|r\|^2 = \|\eta\|^2$ и вследствие (14) условие (12) выполняется точно.

Поскольку функционал невязки $\phi(\gamma)$ является монотонной функцией параметра регуляризации, нахождение искомого значения γ не представляет трудности. Один из алгоритмов состоит в следующем.

1. Задать начальное значение параметра регуляризации γ .

2. Вычислить $\|r\|^2 \leq \|\eta\|^2$.

3. Если условие (15) выполняется, цель достигнута, выходим из цикла. В противном случае необходимо увеличить значение γ , если $\|r\|^2 < \|\eta\|^2 - a$, либо уменьшить его, если $\|r\|^2 > \|\eta\|^2 + a$. Повторить шаг 2, используя новую оценку γ . Для увеличения скорости сходимости можно использовать более продвинутые методы, например метод касательных Ньютона.

Для использования данного алгоритма необходимы величины $\|r\|^2$ и $\|\eta\|^2$. Для вычисления $\|r\|^2$ обратимся к формуле (14) и запишем ее в следующем виде:

$$R(u, v) = G(u, v) - H(u, v) \hat{F}(u, v),$$

причем функция $r(x, y)$ может быть получена вычислением обратного Фурье-преобразования функции $R(u, v)$. Итак,

$$\|r\|^2 = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} r^2(x, y). \quad (16)$$

Рассмотрим дисперсию шума полного изображения

$$\sigma_\eta^2 = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (\eta(x, y) - m_\eta)^2,$$

где $m_{\eta} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \eta(x, y)$ (18) есть среднее значение шума.

Сопоставляя выражения (17), (18) с выражением для $\|\eta\|^2$, которое имеет тот же вид, что и (16), видим, что

$$\|\eta\|^2 = MN(\sigma_{\eta}^2 + m_{\eta}^2).$$

Это важный результат, который показывает, что для реализации оптимального алгоритма восстановления методом минимизации сглаживающего функционала со связью достаточно знать лишь среднее значение шума и его дисперсию.

4. Программная реализация

Разработанные программные обеспечения реализованы в среде *GUI Matlab*.

После запуска приложения для восстановления смазанного изображения появляется главное окно (Рис. 4), где пользователь может выбрать любое изображение и ввести значение параметра регуляризации.

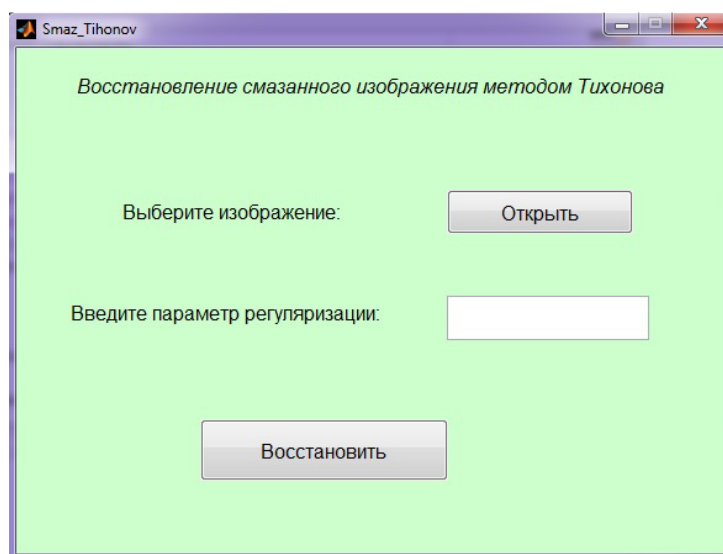


Рис. 4. Главное окно приложения

На Рис. 5-6 представлен графический интерфейс программы, реализующий возможность восстановления изображения с заранее известным параметром регуляризации.

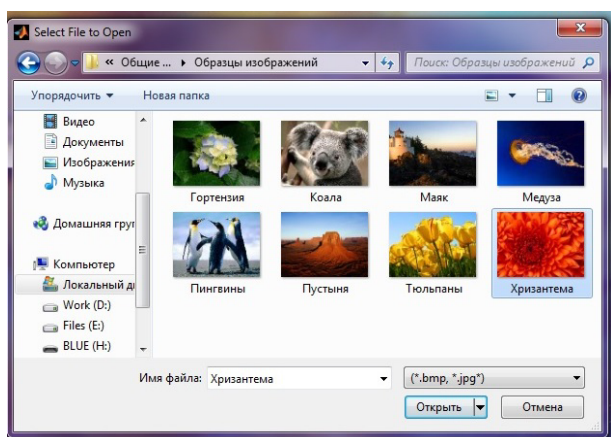


Рис. 5. Выбор изображения

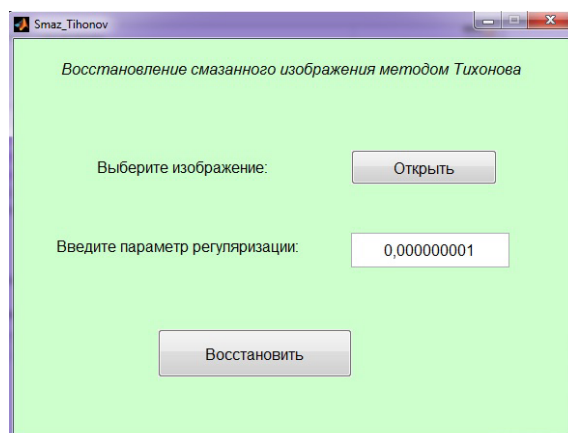


Рис. 6. Ввод параметра регуляризации

На Рис. 7-9 представлен результат работы программы.

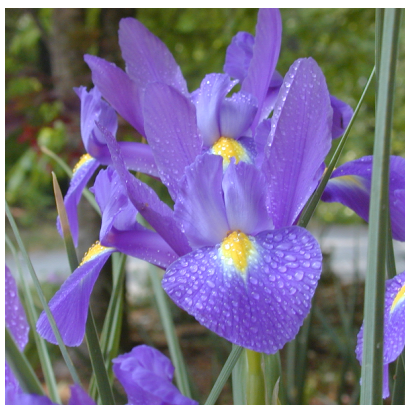


Рис. 7. Исходное изображение

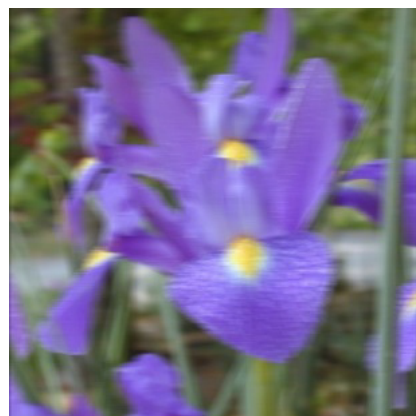


Рис. 8. Смазанное изображение

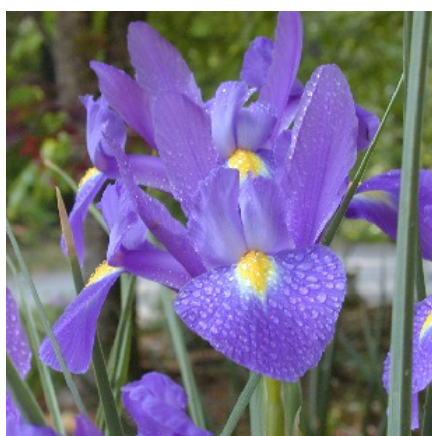


Рис. 9. Восстановленное изображение

После запуска приложения для восстановления расфокусированного изображения появляется главное окно (Рис. 10), где пользователь может выбрать любое изображение и ввести значение параметра регуляризации.

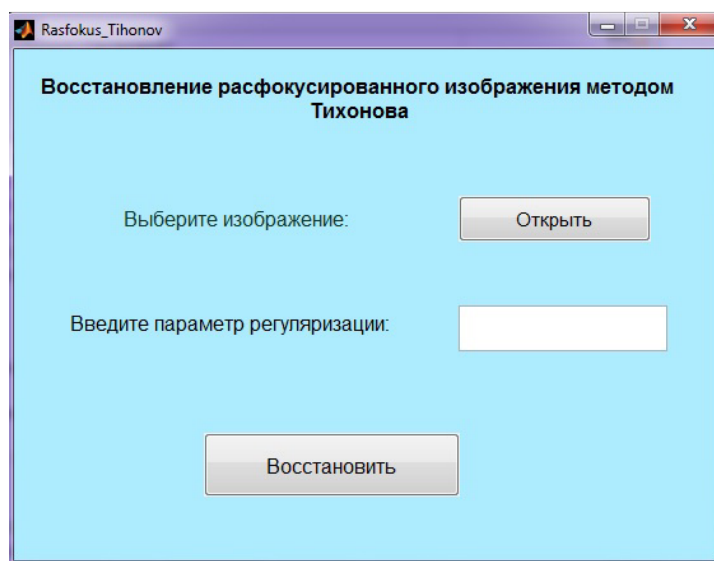


Рис. 10. Главное окно приложения

На Рис. 11-12 представлен графический интерфейс программы, реализующий возможность восстановления изображения с заранее известным параметром регуляризации.

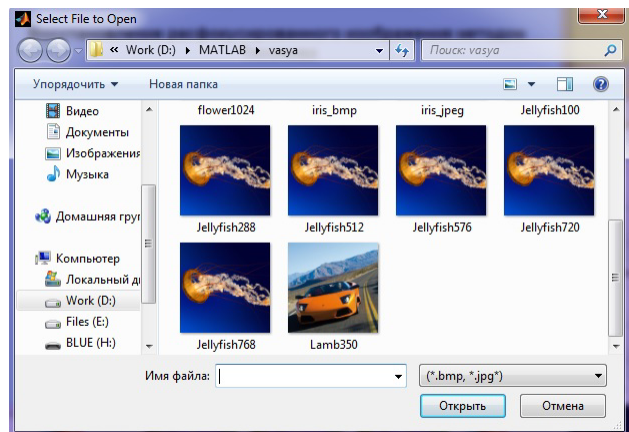


Рис. 11. Выбор изображения

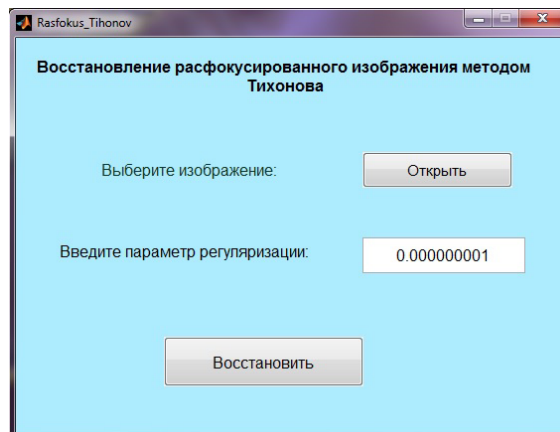


Рис. 12. Ввод параметра регуляризации

На Рис. 13-15 представлен результат работы программы.



Рис. 13. Исходное изображение



Рис. 14. Расфокусированное изображение



Рис. 15. Восстановленное изображение

6. Экспериментальные исследования

Разработанные проекты для оптимизации алгоритма решения задачи восстановления изображения методом Тихонова реализован в среде *Matlab*.

В созданном проекте осуществляется процесс восстановления расфокусированного и смазанного изображения и вычисление времени выполнения функции восстановления. Необходимо определить для каких значений созданный алгоритм работает более устойчиво, а для каких менее.

6.1. Исследование зависимости времени выполнения от размера изображения.

Для экспериментального исследования были выбраны изображения формата *.bmp* различных размеров: 100×100 пикселей, 300×300 пикселей, 512×512 пикселей, 576×576 пикселей, 720×720 пикселей, 768×768 пикселей, 960×960 пикселей, 1024×1024 пикселей.

С помощью созданного приложения проводился анализ зависимости времени выполнения от размера изображения при угле смаза 0° и длине смаза 20 пикселей.

Оценка результатов приведена на графике (Рис. 16). По оси *X* отложены размеры изображения, по оси *Y* - время выполнения.

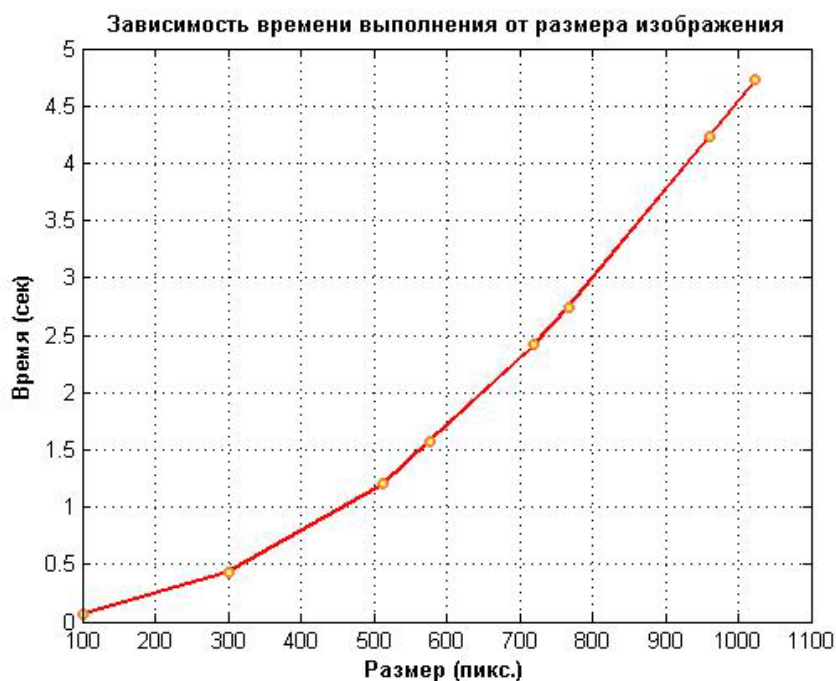


Рис. 16. Зависимость времени выполнения от размера изображения

Из графика (Рис. 16) видно, что при увеличении размера изображения время выполнения возрастает.

6.2. Исследование зависимости СКО от размера изображения

Для экспериментального исследования были выбраны изображения формата *.bmp* различных размеров: 100×100 пикселей, 300×300 пикселей, 512×512 пикселей, 576×576 пикселей, 720×720 пикселей, 768×768 пикселей, 960×960 пикселей, 1024×1024 пикселей.

С помощью созданного приложения проводился анализ зависимости СКО от размера изображения при угле смаза 0° и длине смаза 20 пикселей.

Допустимое значение СКО: 0,1.

Оценка результатов приведена на графике (Рис. 17). По оси *X* отложены размеры изображения, по оси *Y* - СКО.

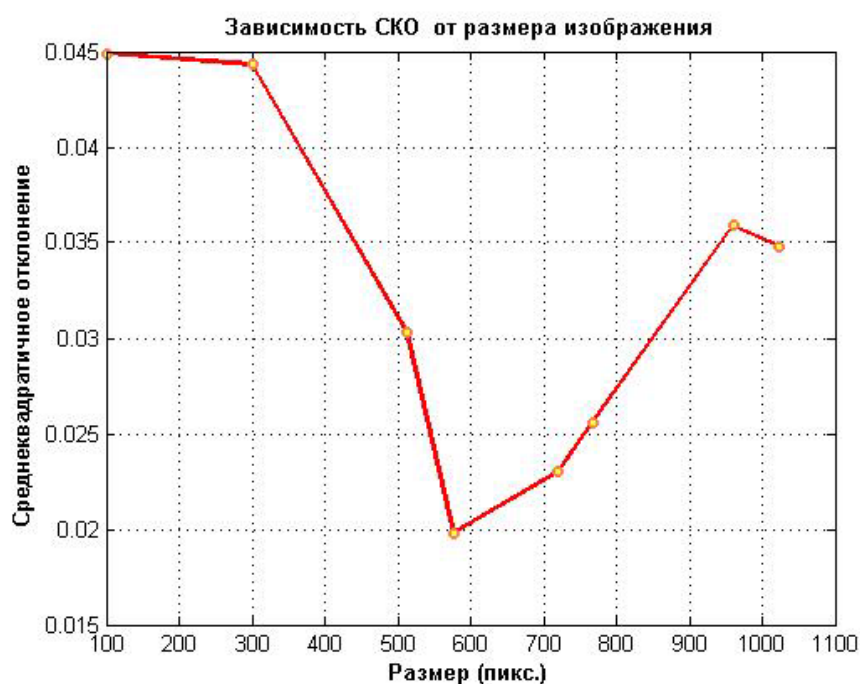


Рис. 17. Зависимость СКО от размера изображения

Из графика (Рис. 17) видно, что при увеличении размера изображения СКО изменяется скачкообразно, но незначительно и находится в пределах допустимого значения.

6.3. Исследование зависимости времени выполнения от формата изображения

Для экспериментального исследования было выбрано изображение размера 300×300 пикселей различных форматов: *.bmp*, *.jpg*, *.png*. С помощью созданного приложения проводился анализ зависимости времени выполнения от формата изображения при угле смаза 0° и длине смаза 20 пикселей.

Оценка результатов приведена на графике (Рис. 18). По оси X отложены форматы изображения ('1'- формат *.bmp*, '2'- формат *.jpg*, '3'- формат *.png*), по оси Y - время выполнения.



Рис. 18. Зависимость времени выполнения от формата изображения

Из графика (Рис. 18) видно, что время выполнения при различных форматах изображения примерно одинаково. Быстрее, чем остальные восстанавливалось изображение формата *.bmp*. Медленнее всех восстанавливалось изображение формата *.png*.

6.4. Исследование зависимости СКО от формата изображения

Для экспериментального исследования было выбрано изображение размера 300×300 пикселей различных форматов: *.bmp*, *.jpg*, *.png*. С помощью созданного приложения проводился анализ зависимости времени выполнения от формата изображения при угле смаза 0° и длине смаза 20 пикселей.

Оценка результатов приведена на графике (Рис. 19). По оси X отложены форматы изображения ('1'- формат *.bmp*, '2'- формат *.jpg*, '3'- формат *.png*), по оси Y - СКО.

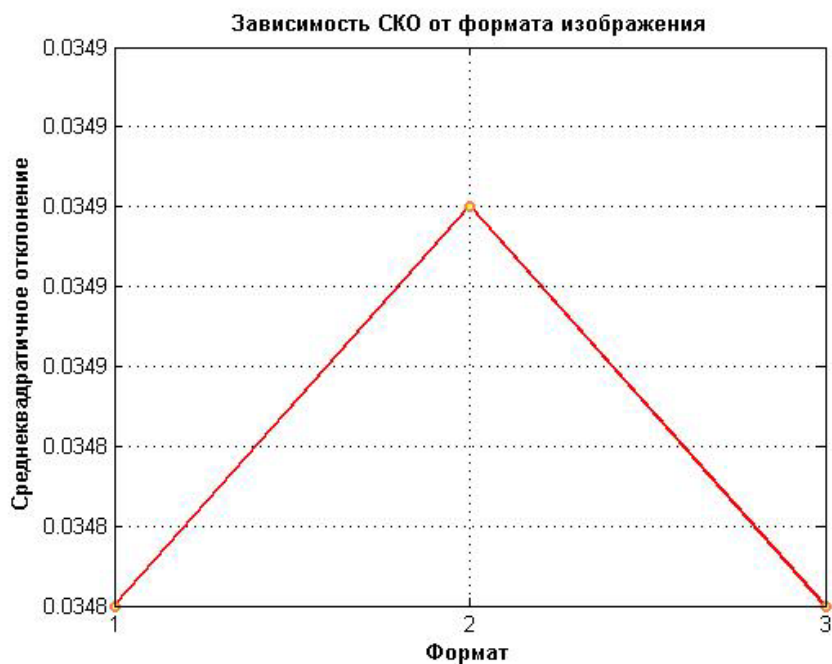


Рис. 19. Зависимость времени выполнения от формата изображения

Из графика (Рис. 19) видно, что при разных форматах изображения СКО изменяется незначительно и находится в пределах допустимого значения.

6.5. Исследование зависимости времени выполнения от параметра регуляризации.

Для экспериментального исследования было выбрано изображение размера 300×300 пикселей формата *.bmp*. С помощью созданного приложения проводился анализ зависимости времени выполнения от параметра регуляризации при угле смаза 0° и длине смаза 20 пикселей.

Оценка результатов приведена на графике (Рис. 20). По оси *X* отложены параметры регуляризации, по оси *Y* - время выполнения.



Рис. 20. Зависимость времени выполнения от параметра регуляризации

Из графика (Рис. 20) видно, что время выполнения меняется скачкообразно с увеличением параметра регуляризации, незначительно возрастает.

6.6. Исследование зависимости СКО от параметра регуляризации

Для экспериментального исследования было выбрано изображение размера 300×300 пикселей формата *.bmp*. С помощью созданного приложения проводился анализ зависимости времени выполнения от параметра регуляризации при угле смаза 0° и длине смаза 20 пикселей.

Оценка результатов приведена на графике (Рис. 21). По оси *X* отложены параметры регуляризации, по оси *Y* - время выполнения.

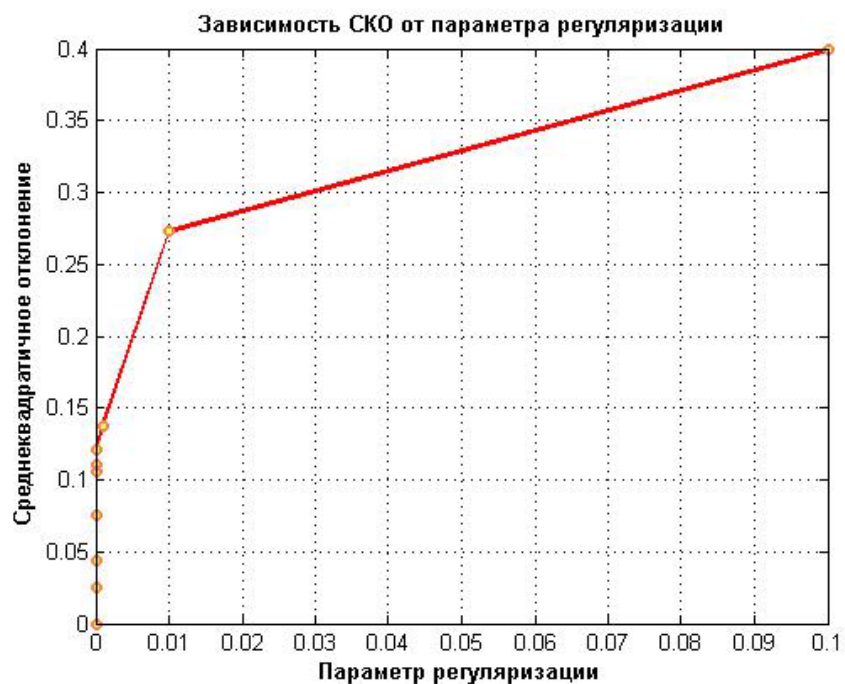


Рис. 21. Зависимость СКО от параметра регуляризации

Из графика (Рис. 21) видно, что с увеличением параметра регуляризации, СКО возрастает.

Причем, в пределах допустимого находятся значения СКО при параметрах регуляризации от $\gamma = 1e-9$ до $\gamma = 1e-7$.

В подтверждение сказанного можно оценить восстановленные изображения при параметрах регуляризации $\gamma = 1e-9$ и $\gamma = 1e-1$ (Рис. 22-27).



Рис. 22. Расфокусированное изображение



Рис. 23. Восстановленное изображение с параметром регуляризации $1e-9$



Рис. 24. Восстановленное изображение с параметром регуляризации $1e-1$

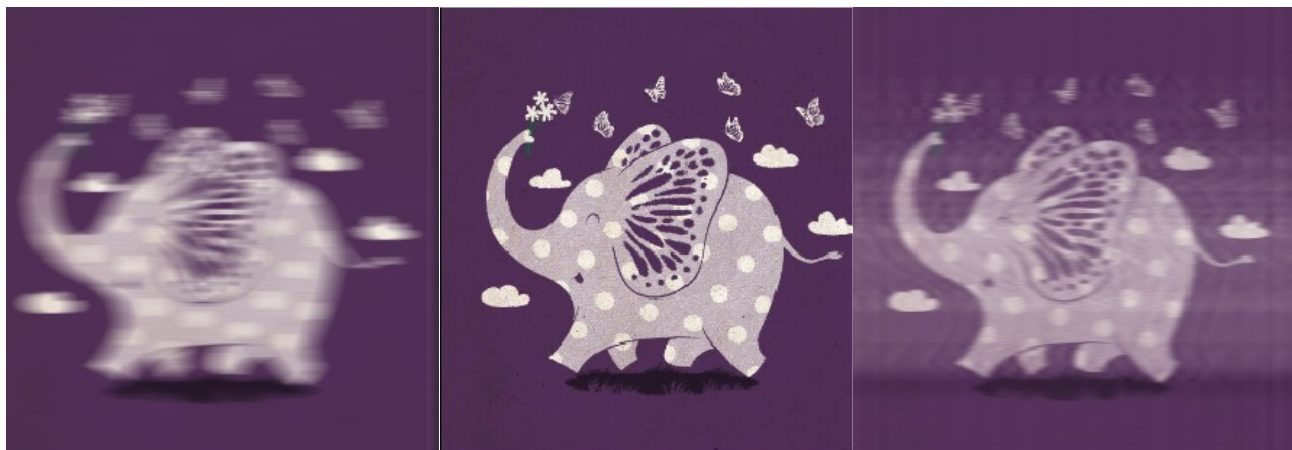


Рис. 25. Смазанное изображение

Рис. 26. Восстановленное изображение с параметром регуляризации $1e-9$

Рис. 27. Восстановленное изображение с параметром регуляризации $1e-1$

Очевидно, что наилучшее качество изображения достигается при параметре регуляризации $\gamma = 1e-9$. Большие значения параметра регуляризации не являются подходящими для качественного восстановления расфокусированного изображения методом Тихонова.

7. Заключение

В результате проведения данной работы было разработано программное приложение для оптимизации алгоритма восстановления смазанного изображения методом регуляризации Тихонова. Достоинством данного приложения является автоматизация процесса восстановления изображения. Значительных недостатков приложения выявлено не было. В рамках данного проекта поставленные задачи были выполнены.

Список литературы

- [1] Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений. 3-е изд., перераб. и доп. М.: Техносфера, 2012. 1101с.
- [2] Волосатова Т.М., Воронов А.В., Чичварин Н.В. Математическое и программное обеспечение решения задачи синтеза для САПР оптико-электронных систем. Информационные технологии: Приложение к журналу. 2008. № 4. С. 2 - 32.
- [3] Волосатова Т.М., Денисов А.В., Чичварин Н.В. Комбинированные методы защиты

данных в САПР // Информационные технологии. 2012. № 5. Приложение. С. 1 - 32.

- [4] Винер Н. Нелинейные задачи в теории случайных процессов. М.: Книга по Требованию, 2012. 158 с.
- [5] Журавель И.М. Краткий курс теории обработки изображений. Консультационный центр MATLAB компании *Softline*. Режим доступа: <http://matlab.exponenta.ru/imageprocess/book2/76.php> (дата обращения 20.04.2015).