

УДК 519.642.3

ВОССТАНОВЛЕНИЕ СМАЗАННЫХ И ДЕФОКУСИРОВАННЫХ ЦВЕТНЫХ ИЗОБРАЖЕНИЙ

А. Г. Ягола¹, Н. А. Кошев¹

Задача восстановления изображений, подвергшихся аддитивным пространственно-инвариантным искажениям, сводится к решению интегрального уравнения типа свертки или системы таких уравнений. Решение проводится с применением алгоритмов регуляризации некорректно поставленных задач. Приводится описание метода, основанного на минимизации функционала Тихонова, описание программы и примеры ее работы (восстановленные изображения). Работа поддержана грантами РФФИ 08-01-00160-а и 07-01-92103-ГФЕН-а.

Ключевые слова: восстановление изображений, дефокусированные изображения, смазанные изображения, метод регуляризации Тихонова, интегральные уравнения, уравнения типа свертки.

1. Постановка задачи. Актуальность обратных задач восстановления цифровых изображений обусловлена быстрым развитием вычислительной техники в последние годы. Такие задачи, исследованные еще в 70-е годы прошлого века, обладают огромной ресурсоемкостью, вызванной многократным применением интегральных преобразований (преобразований Фурье, Лапласа и т.п.) к большим численным массивам. Несмотря на развитие оптимизационных алгоритмов (например, быстрое преобразование Фурье), лишь в настоящее время появилась возможность обработки изображений высокого разрешения на стандартных компьютерах без привлечения значительных вычислительных ресурсов. В связи с этим интерес к задаче возникает как у научных организаций (астрономических, медицинских, геологических и т.д.), так и у коммерческих. Например, знание характеристик объектива фотоаппарата позволяет построить аппаратную функцию для данного аппарата и восстанавливать изображения, снятые при его помощи, с использованием обычных компьютеров.

Конечно, при этом возникает большое количество трудностей, нуждающихся на данный момент в изучении и разрешении. Настоящая статья представляет собой несколько упрощенное рассмотрение задачи восстановления цифровых изображений. Однако даже такая упрощенная формулировка задачи, как мы увидим далее, вполне пригодна для некоторого класса реальных задач, удовлетворяющих следующим предположениям:

- 1) объектив камеры представляет собой тонкую линзу;
- 2) снимаемый объект лежит в плоскости, перпендикулярной главной оптической оси;
- 3) размер объекта гораздо меньше радиуса апертуры объектива;
- 4) зафиксированная матрицей камеры интенсивность будем считать равной интенсивности падающего света.

При рассмотрении задачи в контексте описанных допущений можно пренебречь возникновением нелинейных оптических эффектов, что значительно упрощает задачу.

Смазывание и дефокусировка являются аддитивными пространственно-инвариантными искажениями. Пусть z — распределение интенсивностей неискаженного снимка, u — распределение интенсивностей в кадре, k — аппаратная функция искажающей системы. Тогда u выразится через z посредством свертки (см. [3]):

$$u = k z. \tag{1}$$

Согласно теореме о свертке, после применения преобразования Фурье к правой и левой частям уравнения (1) получим уравнение $U = KZ$, где $U = \mathcal{F}[u]$, $K = \mathcal{F}[k]$, $Z = \mathcal{F}[z]$ и $X(\lambda) \equiv \mathcal{F}[x] \equiv \int_{-\infty}^{\infty} x(t)e^{-it\lambda} dx$ — прямое преобразование Фурье.

Решение уравнения (1) нельзя искать в виде $z = \mathcal{F}^{-1} \left[\frac{U}{K} \right]$, где $x(t) \equiv \mathcal{F}^{-1}[x] \equiv \int_{-\infty}^{\infty} X(t)e^{it\lambda} d\lambda$ —

¹ Московский государственный университет имени М. В. Ломоносова, физический факультет, 119991, Ленинские горы, Москва; e-mail: yagola@inverse.phys.msu.ru, koshev@mail.ru

обратное преобразование Фурье, так как в силу свойств преобразования Фурье такое решение не является устойчивым [1, 2] относительно u .

1.1. Задача восстановления смазанного изображения. Смазанное изображение возникает вследствие перемещения объекта и камеры относительно друг друга за время выдержки (время, когда диафрагма камеры открыта). Без ограничения общности можно положить, что это перемещение вызвано движением объекта относительно камеры. Здесь мы считаем перемещение параллельным одной из координатных осей.

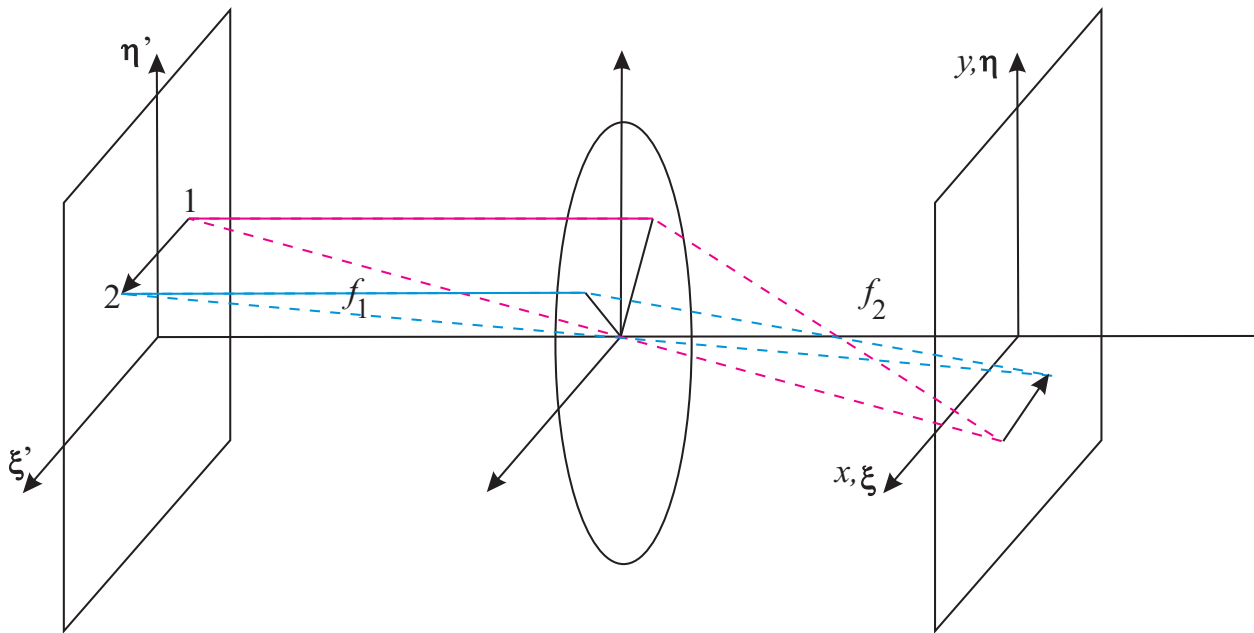


Рис. 1. Схема образования смазанного изображения

На рис. 1 использованы следующие обозначения: (ξ', η') — система координат, заданная в плоскости объекта; (ξ, η) и (x, y) — совпадающие системы координат в плоскости изображения; f_1 и f_2 — расстояния от объекта до объектива и от объектива до экрана соответственно. Цифрами 1 и 2 обозначены положения объекта относительно камеры в моменты открытия и закрытия диафрагмы соответственно [4, 5]. Задача сводится к решению семейства уравнений

$$u(x, y) = \int_{-\infty}^{\infty} k(x - \xi)z(\xi, y) d\xi, \quad (2)$$

где $z(\xi, y)$ — идеальное (несмазанное) изображение, $u(x, y)$ — реальное изображение, полученное на экране, l — расстояние между положениями 1 и 2 (рис. 1) и $k(x)$ — аппаратная функция:

$$k(x) = \begin{cases} \frac{1}{2l}, & |x| < l, \\ 0, & |x| \geq l. \end{cases} \quad (3)$$

Уравнения (2) представляют собой однопараметрическое семейство интегральных уравнений Фредгольма первого рода. Задача (2), (3) является некорректной в силу неустойчивости решения относительно малых изменений u . Для нахождения распределения интенсивностей идеального изображения воспользуемся регуляризирующим алгоритмом Тихонова [1] ($\alpha > 0$ — параметр регуляризации):

$$z_{\alpha}(\xi, y) = \int_{-\infty}^{\infty} \frac{K^*(\lambda)U(\lambda, y)}{L(\lambda) + \alpha M(\lambda)} e^{-i\lambda\xi} d\lambda. \quad (4)$$

В дискретном случае выражение (4) принимает вид

$$z_{kj} = \sum_{n=-N/2}^{n=N/2-1} \frac{K_n^* U_{nj}}{K_n^* K_n + \alpha (nd)^{2p}} e^{-id^2 nk}, \quad (5)$$

где N — количество узлов сетки (пикселей в изображении), d — шаг сетки, z_{kj} — двумерный массив распределения интенсивности неискаженного изображения, U_{nj} — двумерный массив распределения интенсивности полученного (восстанавливаемого) изображения и K_n — одномерный массив, содержащий значения аппаратной функции.

Итак, задача сводится к нахождению всех элементов массива дискретного распределения интенсивностей (5).

1.2. Задача восстановления дефокусированного изображения. Рассмотрим далее аналогичный алгоритм для восстановления дефокусированных изображений.

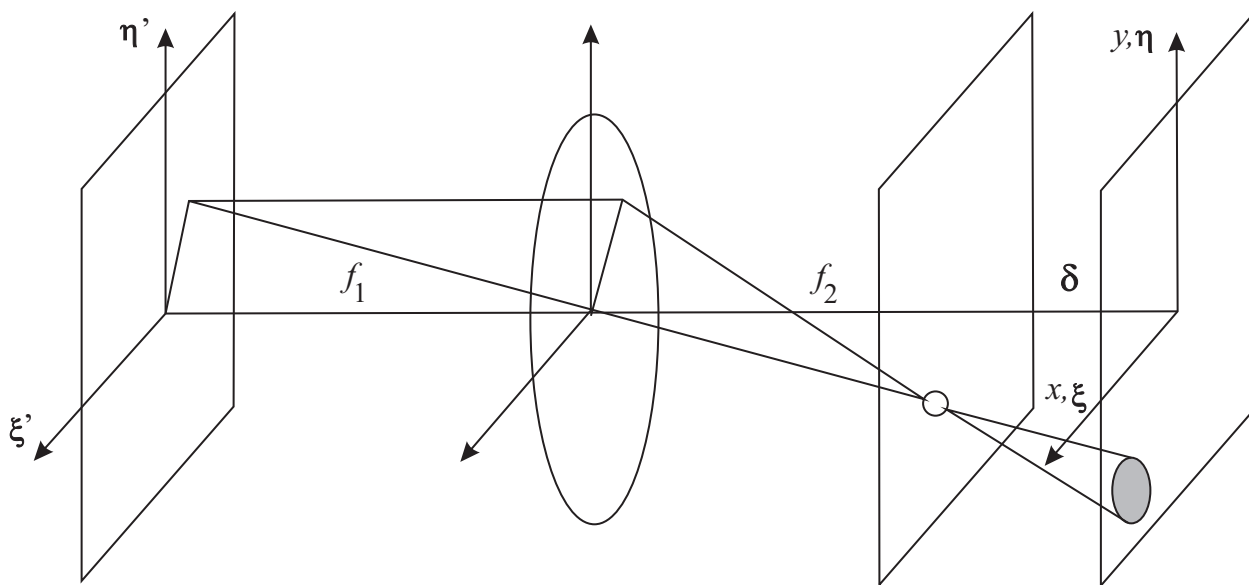


Рис. 2. Схема формирования дефокусированного изображения

На рис. 2, как и для предыдущего случая, f_1 и f_2 — расстояния от объекта до объектива и от объектива до экрана соответственно; (ξ', η') — система координат, заданная в плоскости объекта; (ξ, η) и (x, y) — совпадающие системы координат в плоскости изображения; δ — погрешность фокусировки (расстояние между матрицей и фокальной плоскостью объектива) [4, 5].

Обозначая, как и ранее, через $z(\xi, \eta)$ распределение для неискаженного изображения (находящегося в фокальной плоскости), а через $u(x, y)$ — для реального, выпишем закон построения (соответственно, и задачу восстановления) дефокусированного изображения:

$$u(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k(x - \xi, y - \eta) z(\xi, \eta) d\xi d\eta, \tag{6}$$

где

$$k(x, y) = \begin{cases} \frac{1}{\pi\rho^2}, & x^2 + y^2 \leq \rho^2, \\ 0, & x^2 + y^2 > \rho^2 \end{cases}. \tag{7}$$

Заметим, что уравнение (6) является двумерным, в то время как задача (2), (3) представляет собой однопараметрическое семейство одномерных уравнений. Это связано с двумерностью аппаратной функции в случае дефокусировки. Следует отметить, что для случая смазывания аппаратная функция является одномерной только при линейном смещении объекта относительно камеры.

На рис. 3 изображено ядро и модуль его преобразования Фурье. Запишем далее регуляризованное решение задачи (6), (7):

$$R_\alpha[u, \alpha] = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{K^*(\lambda, v)U(\lambda, v)}{L(\lambda, v) + \alpha(\lambda^2 + v^2)^p} e^{-i(x\lambda + yv)} d\lambda dv. \tag{8}$$

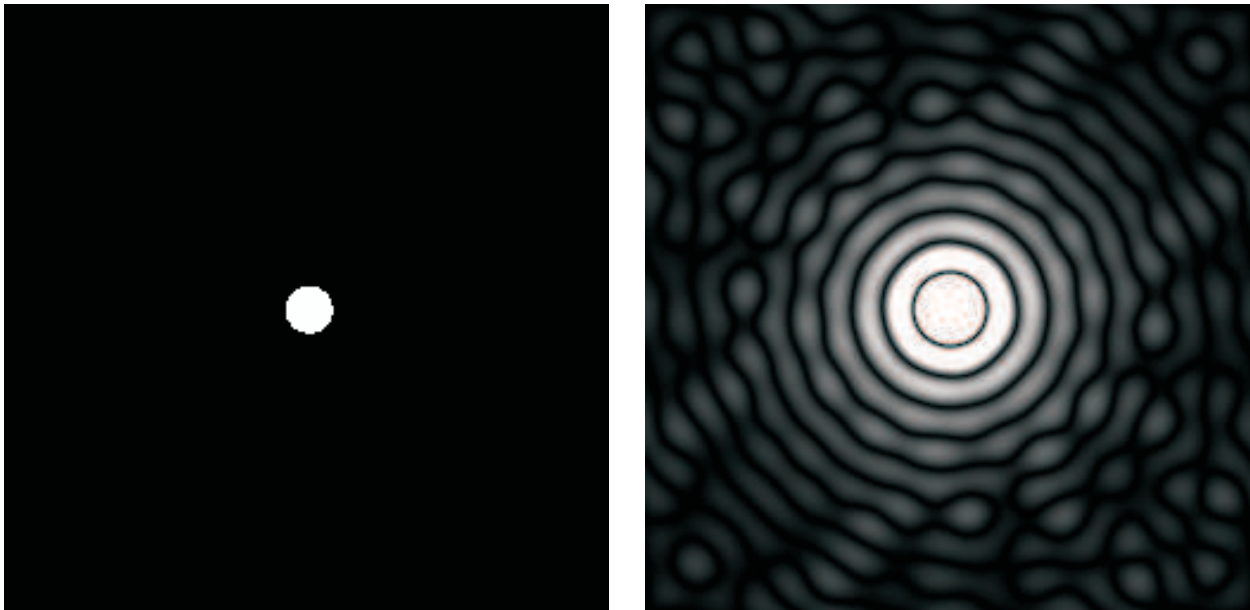


Рис. 3. Ядро и модуль его преобразования Фурье

Здесь $\alpha > 0$ — параметр регуляризации и $L(\lambda, v) = |K(\lambda, v)|$. В дискретном случае имеем

$$z_{lj} = \frac{1}{d_1 d_2} \sum_{n=0}^{n=N_1} \sum_{k=0}^{k=N_2} \frac{K_{nk}^* U_{nk}}{K_{nk}^* K_{nk} + \alpha \pi^2 (n^2 + k^2) p d^4} e^{-i\pi(\ln d_1^2 + jk d_2^2)}, \quad (9)$$

где z_{lj} — двумерный массив распределения интенсивности неискаженного изображения, U_{nk} — двумерный массив распределения интенсивности полученного (восстанавливаемого) изображения, d_1 и d_2 — шаги сетки, N_1 и N_2 — количество узлов сетки и K_{nk} — двумерный массив, содержащий значения аппаратной функции.



Рис. 4. Смазанное изображение. Модельная задача: а) смазанная фотография, б) восстановленная фотография. Разрешение — 1024×768 , время расчета — 8 секунд

2. Краткое описание программы. В случае обработки цветного изображения цвет каждого дискрета (пикселя) раскладывается по базису RGB (red-green-blue). Для каждого цвета составляется массив распределения интенсивностей. Для 24-битовых изображений интенсивность каждого цвета меняется в пределах $(0, 255)$, что обеспечивает около 17 миллионов цветов. После составления массивов каждый из них обрабатывается отдельно и поочередно. Очередность желательна для уменьшения ресурсоемкости

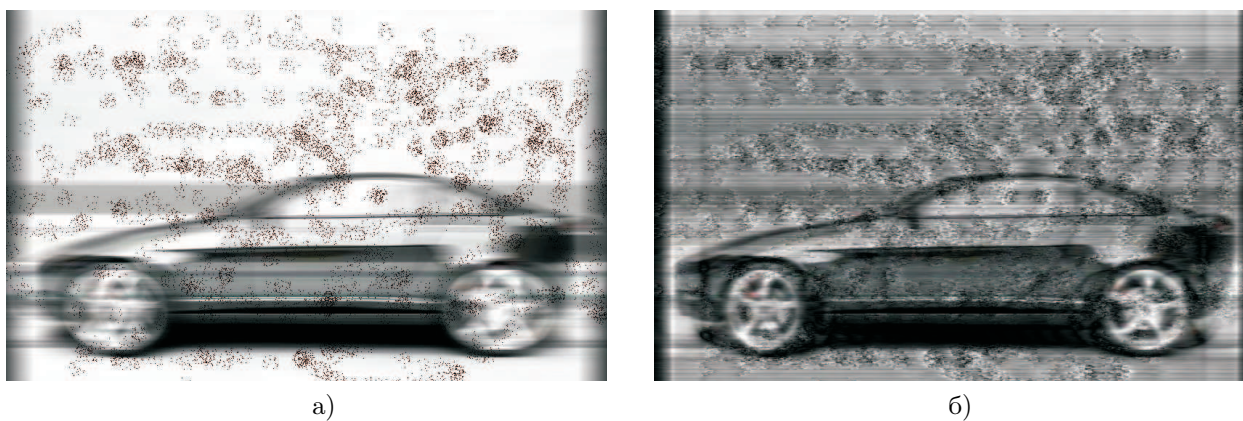


Рис. 5. Смазанное изображение с добавлением шума: а) искусственно зашумленная смазанная фотография, б) восстановленная фотография



Рис. 6. Дефокусированное изображение. Модельная задача: а) дефокусированная фотография, б) восстановленная фотография. Разрешение — 1024×768 , время расчета — 11 секунд



Рис. 7. Реальная задача. Восстановление изображения на заднем плане: а) дефокусированная фотография, б) восстановленная фотография. Разрешение — 1024×768 , параметры регуляризации по всем цветам одинаковы и равны 0.007. ФРТ: $\rho = 9$ точек

задачи, но, конечно же, приводит к увеличению временных затрат. На многоядерных процессорах или вычислительных кластерах удобнее применять параллельные вычислительные потоки для расчета каждого цвета одновременно.

Программа написана в среде C++ с применением библиотеки QT-4.3.2 [6]. Для повышения производительности использовались “вытянутые” массивы и адаптированное для них быстрое преобразование

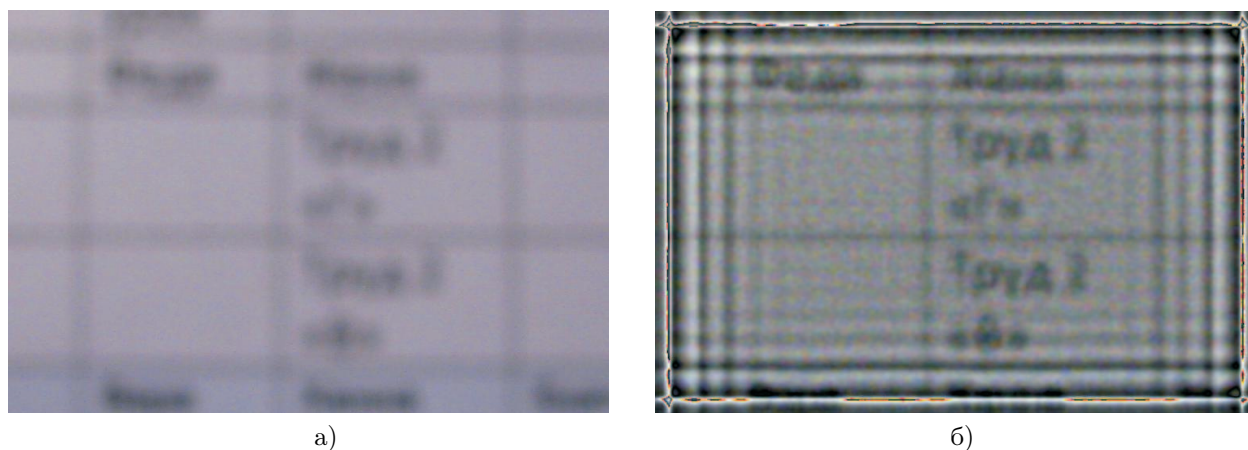


Рис. 8. Реальная задача. Восстановление сильно дефокусированного текста: а) дефокусированная фотография, б) восстановленная фотография. Разрешение — 1024×768 , параметры регуляризации по всем цветам одинаковы и равны 0.01. ФРТ: $\rho = 19$ точек

Фурье (БПФ) [7].

Рисунки 4–8 иллюстрируют примеры восстановления изображений с использованием нашей программы. Расчеты проводились на компьютере AMD Athlon 64 X2 Dual Core Processor 3800+, оперативная память 2 Gb, тактовая частота каждого ядра — 2 MHz.

Из рис. 5 следует, что рассмотренный алгоритм довольно устойчив по отношению к шумам.

Отметим, что выбор ФРТ (функция размытия точки), так же как и выбор параметра регуляризации, остается за пользователем и осуществляется им на основании ожидаемого визуального качества восстановленного изображения. Следует также отметить хорошие результаты для модельных фотографий на темном фоне (рис. 4 и 6), что делает возможным использование данного приложения в задачах астрономии.

Нами использовалась тонкая линза в качестве модели объектива камеры, однако аппаратная функция в программе легко заменяется на любую другую.

СПИСОК ЛИТЕРАТУРЫ

1. Тихонов А.Н., Гончарский А.В., Степанов В.В., Ягола А.Г. Численные методы решения некорректных задач. М.: Наука, 1990.
2. Тихонов А.Н., Арсенин В.Я. Методы решения некорректных задач. М.: Наука, 1990.
3. Грузман И.С., Киричук В.С., Косых В.П. и др. Цифровая обработка изображений в информационных системах. Новосибирск: НГТУ, 2000.
4. Тихонов А.Н., Гончарский А.В., Степанов В.В., Кочкиков И.В. Некорректные задачи обработки изображений // Докл. АН СССР. 1987. **294**, № 4. 832–837.
5. Тихонов А.Н., Гончарский А.В., Степанов В.В. Обратные задачи обработки фотоизображений // Некорректные задачи естествознания / Под ред. Тихонова А.Н., Гончарского А.В. М.: Изд-во Моск. ун-та, 1989. 185–195.
6. <http://trolltech.com/>
7. Арушанян О.Б., Волченкова Н.И. Библиотека программ НИВЦ МГУ для решения типовых задач численного анализа (<http://num-anal.srcc.msu.ru>) // Вычислительные методы и программирование. 2002. **3**, № 2. 158–163.

Поступила в редакцию
02.06.2008