УДК 621.397.3: 517.968

ПРИЕМ “УСЕЧЕНИЕ–РАЗМЫТИЕ–ПОВОРОТ” ДЛЯ ВОССТАНОВЛЕНИЯ ИСКАЖЕННЫХ ИЗОБРАЖЕНИЙ

В. С. Сизиков, доктор техн. наук

Санкт-Петербургский государственный университет информационных технологий, механики и оптики, Санкт-Петербург

E-mail: sizikov2000@mail.ru

Рассматривается задача восстановления искаженных (смазанных, дефокусированных, зашумленных) серых и цветных изображений. Устранение смазывания и дефокусирования изображений выполняется путем решения интегральных уравнений методом регуляризации Тихонова или параметрической фильтрации Винера, а устранение шума – методом адаптивной фильтрации Винера или медианной фильтрации. Вместо так называемых “граничных условий” предлагается обобщенный прием усечения изображения, а для понижения эффекта Гиббса – обобщенный прием размытия краев изображения. Для моделирования смазывания изображения под произвольным углом предложен прием поворота изображения. Методика реализована в виде m-файлов в системе MatLab. Выполнена обработка модельных и натурных изображений.

Ключевые слова: *смазывание, дефокусирование, зашумление изображения, “граничные условия”, усечение, размытие, поворот изображения.*

Коды OCIS: 100.0100, 100.2000

Поступила в редакцию 02.11.2010

Введение

Данная работа является продолжением работ [1–6]. Рассматривается задача восстановления (реконструкции) искаженных (смазанных, дефокусированных, зашумленных) изображений. Решается как прямая задача (моделирование искажения), так и обратная задача (восстановление изображения) [1–15]. Основные соотношения, описывающие эти задачи, имеют вид

$$\int_{-\infty}^{\infty} h(x-\xi)w_y(\xi)d\xi = g_y(x)+\delta g, \qquad (1)$$

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x-\xi, y-\eta)w(\xi,\eta)d\xi d\eta = g(x,y)+\delta g, \qquad (2)$$

где h – функция рассеяния точки (ФРТ, PSF, аппаратная функция), обычно пространственно-инвариантная, или разностная, w и g – распределение интенсивности по неискаженному и искаженному изображениям соответственно, δg – помеха (шум). В (1) ось x направлена вдоль смаза, а y играет роль параметра.

Набор одномерных интегральных уравнений (ИУ) (1) обычно используется в задаче смазывания, а двумерное ИУ (2) – в задаче дефокусирования, но часто [9, 11] ИУ (2) используется для решения обеих задач или смешанных задач [12, 16, 17], когда ФРТ имеет произвольную форму.

При решении прямой и обратной задач часто используются [9, 10, 16–20] так называемые “граничные условия” (ГУ) для аппроксимации неизвестных (но нужных) значений w вне границ g поля зрения (FOV – field of view [16, 18]), когда функция w не является финитной, т. е. носитель изображения w шире FOV. Однако более точным следует считать термин “внеграничные условия”.

Непрерывным уравнениям (1), (2) соответствует дискретное соотношение

$$Aw = g+\delta g, \qquad (3)$$

где A – сложная матрица [18], связанная с ФРТ и с типом ГУ.

В работах [1, 5] предложен прием усечения изображения g, позволяющий избежать использования ГУ. Предложен также прием размытия краев изображения g, чтобы понизить эффект Гиббса (эффект ложных волн) на восстанавливаемом изображении w. Однако в работах [1, 5] прием усечения–размытия развит применительно лишь к смазыванию изображения, причем под нулевым углом (горизонтальное смазывание) и для серого изображения. В данной работе этот прием распространяется на смазывание под произвольным углом и на дефокусирование, как для серого, так и цветного (RGB) изображения. Будем называть такой прием обобщенным приемом усечения–размытия.

Обработка изображения, смазанного под произвольным углом

Прямая задача. Сначала рассмотрим случай, когда угол смазывания θ равен нулю (горизонтальное смазывание). При этом нижеследующие соотношения будем излагать, в основном, применительно к системе MatLab. В работах [1, 5] предложен прием усечения смазанного изображения, согласно которому

$$\hat{g}_j(i) = \frac{1}{\Delta+1}\sum_{k=i}^{i+\Delta} w_j(k), \quad i = 1, 2, \ldots, n-\Delta, \quad j = 1, 2, \ldots, m, \tag{4}$$

где i (и k) – номер столбца, j – номер строки изображения, $m\times n$ – размер матрицы исходного изображения w, $m\times(n–\Delta)$ – размер матрицы моделируемого смазанного изображения $\hat{g}$, Δ – величина смаза.

Чтобы при решении обратной задачи понизить возможные искажения типа “звона” (ложные волны, эффект Гиббса) [10, с. 185], в работе [1] предложен прием размытия краев у изображения g, согласно которому

$$g_j(i) = \frac{1}{\Delta+1}\sum_{k=i}^{i+\Delta} q_j(k), \quad i = 1, 2, \ldots, n+\Delta, \quad j = 1, 2, \ldots, m, \tag{5}$$

где

$$q_j(k) = \begin{cases} w_j(k-\Delta), & k-\Delta \in [1, n], \\ 0, & k-\Delta \notin [1, n]. \end{cases} \tag{6}$$

Однако соотношения (5)–(6) неудобны для программирования в системе MatLab. Можно запись (5)–(6) сделать более удобной, дополнив массив w слева и справа нулевыми матрицами размера $m\times\Delta$ (в MatLab'е это можно выполнить с помощью функции padarray.m [10]). Полученный массив обозначим через u. Тогда соотношения (5)–(6) запишутся в более удобном виде

$$g_j(i) = \frac{1}{\Delta+1}\sum_{k=i}^{i+\Delta} u_j(k), \quad i = 1, 2, \ldots, n+\Delta, \quad j = 1, 2, \ldots, m. \tag{7}$$

Теперь рассмотрим случай произвольного угла смазывания θ. В системе MatLab7 в m-функции fspecial.m [10] заложен численный алгоритм смазывания под произвольным углом, в котором для вычисления разностной ФРТ $h(x, y)$ (см. (2)) учитываются пикселы, отстоящие от каждой наклоненной под углом θ прямой линии не более, чем на 1 пиксел. При этом используется билинейная интерполяция. В результате ФРТ получается в виде матрицы (ве́ктора под углом). Далее в функции imfilter.m [10] для получения смазанного изображения $g(x, y)$ выполняется свертка истинного изображения $w(x, y)$ с ФРТ $h(x, y)$

$$g(x, y) = \iint_D h(x-\xi, y-\eta)w(\xi, \eta)d\xi d\eta, \quad x, y \in D, \tag{8}$$

где D – область изображения w.

Особенности алгоритма, заложенного в функциях fspecial.m и imfilter.m: задача смазывания, как и задача дефокусирования, решается как 2-мерная задача; используются “граничные условия”. Однако, как и в работах [1, 5, 6], предлагается рассматривать задачу смазывания как набор 1-мерных задач (см. (1)), а вместо ГУ использовать обобщенные приемы усечения и размытия краев. При этом вместо учета пикселов, близких к наклонной прямой линии, заложенного в функции fspecial.m, предлагаем использовать прием поворота изображения, который можно выполнить с помощью функции imrotate.m [10].

Изложим суть приема поворота с размытием на примере рис. 1. Исходное изображение w (textp.bmp, рис. 1а) поворачивается на некоторый угол θ с помощью функции imrotate.m. При этом функция imrotate.m заполняет нулями область вне повернутого изображения (рис. 1б). Повернутый массив дополняем слева и справа нулями по Δ пикселов с помощью функции padarray.m, чтобы получить массив u. В заключение рассчитываем значения g с размытием по формуле (7) (рис. 1б) или с усечением по формуле (4).

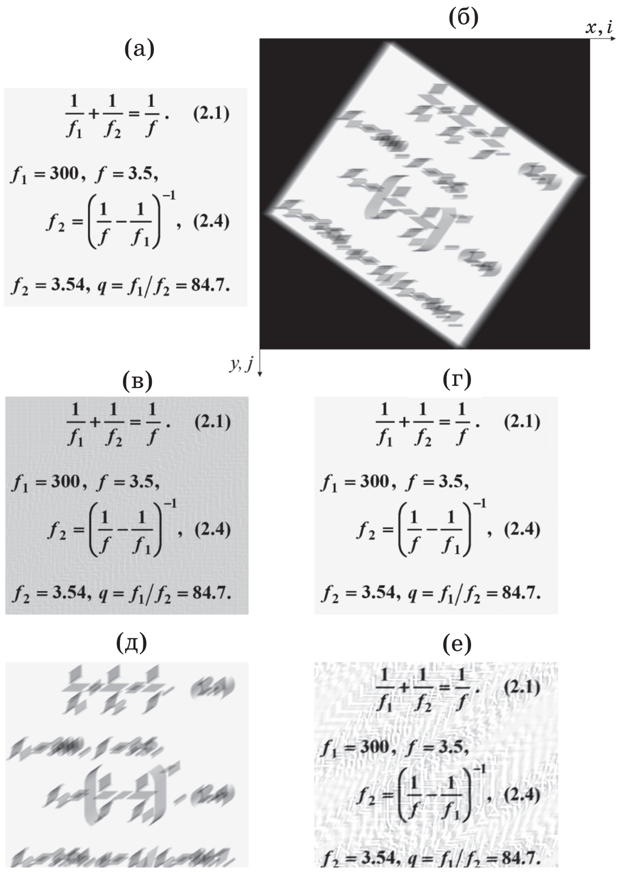

Рис. 1. Прямая и обратная задачи обработки смазанного под углом серого текстового изображения textp.bmp ($\Delta = 40$ пикселов, $\theta = 35°$, без шума). а – исходное изображение w 618×690, б – повернутое и смазанное изображение с размытием краев g 902×960, в – реконструированное изображение w_α, вариант 2, г – реконструированное изображение w_α, вариант 4, д – изображение, смазанное с помощью функций fspecial.m и imfilter.m, е – реконструированное изображение w_K.

Обратная задача. Повернутое изображение (g на рис. 1б) реконструируется в каждой j-й строке путем решения ИУ (1). Как и в работах [1–6], это решение будем выполнять методом ПФ с регуляризацией Тихонова (варианты 1 и 2 [1]) и методом квадратур с регуляризацией Тихонова (варианты 3–5 [1]). После получения решения w_α выполняем обратный поворот изображения и удаляем черную рамку вокруг него (возникающую от функций imrotate.m и padarray.m).

Параметр регуляризации α в методе регуляризации Тихонова выбираем двумя способами: путем визуальной оценки реконструированного изображения w_α и/или путем минимизации относительного среднеквадратического отклонения (СКО) изображения w_α от исходного точного изображения $\bar{w}$ [1]

$$\sigma_{\rm rel} = \frac{\|w_\alpha - \bar{w}\|_{L_2}}{\|\bar{w}\|_{L_2}} = \frac{\sqrt{\sum_{j=1}^{m}\sum_{i=1}^{n}\left[(w_\alpha)_{ji} - \bar{w}_{ji}\right]^2}}{\sqrt{\sum_{j=1}^{m}\sum_{i=1}^{n}\bar{w}_{ji}^2}} \qquad (9)$$

($\sigma_{\rm rel}$ можно вычислить лишь для модельного изображения, когда $\bar{w}$ известно). Обозначим α, соответствующее минимуму $\sigma_{\rm rel}$, через $\alpha_{\rm opt}$.

На рис. 1в приведено изображение w_α, реконструированное методом ПФ и регуляризации Тихонова, вариант 2, $\alpha = \alpha_{\rm opt} = 10^{-6}$, $p = 1$ (порядок регуляризации), $\sigma_{\rm rel} = 0{,}1046$, а на рис. 1г – изображение, реконструированное методом квадратур и регуляризации Тихонова, вариант 4, $\alpha = \alpha_{\rm opt} = 10^{-6}$, $\sigma_{\rm rel} = 0{,}0323$.

Для сравнения на рис. 1д приведено изображение, смазанное с помощью функций fspecial.m и imfilter.m с опцией ("граничным условием") 'circular' (periodic), и на рис. 1е – изображение w_K, реконструированное методом параметрической фильтрации Винера с помощью функции deconvwnr.m [9, 10] с константой (параметром) $K = 10^{-5}$, выбранной, как и α, из условия минимума $\sigma_{\rm rel}$ (в (9) вместо w_α нужно положить w_K), $\sigma_{\rm rel} = 0{,}0789$.

Видим, как и в работе [1], что вариант 4 (метод квадратур с регуляризацией Тихонова и с размытием краев изображения) дает наиболее точное восстановление изображения (без шума).

Зашумление смазанного изображения. Если помимо смазывания изображение зашумлено, то задачу восстановления можно решать двумя путями: использовать метод регуляризации Тихонова (или метод параметрической фильтрации Винера), полагая, что этот метод не только устранит смазывание, но и отфильтрует шум; предварительно отфильтровать шум некоторым специальным методом (адаптивным фильтром Винера или медианным фильтром [21]), а затем применить метод регуляризации Тихонова (или другой метод).

Такого рода численный эксперимент был выполнен применительно к смазанному текстовому изображению, представленному на рис. 1б. Был добавлен гауссовый 1%-ный шум. Согласно 1-му пути, были получены следующие результаты: (в) $\alpha_{\rm opt} = 10^{-2}$, $p = 1$, $\sigma_{\rm rel} = 0{,}1858$; (г) $\alpha_{\rm opt} = 2{\times}10^{-3}$, $\sigma_{\rm rel} = 0{,}0942$; (е) $K_{\rm opt} = 10^{-4}$, $\sigma_{\rm rel} = 0{,}0879$. Согласно 2-му пути (с предварительной фильтрацией шума адаптивным фильтром Винера с помощью m-функции wiener2.m [10, 21]), (в) окно 5×5, $\alpha_{\text{opt}} = 0{,}7\times10^{-2}$, $p = 1$, $\sigma_{\text{rel}} = 0{,}1741$; (г) окно 3×3, $\alpha_{\text{opt}} = 0{,}8\times10^{-3}$, $\sigma_{\text{rel}} = 0{,}0831$; (е) окно 3×3, $K_{\text{opt}} = 0{,}8\times10^{-4}$, $\sigma_{\text{rel}} = 0{,}0900$. Видим, что предварительная фильтрация шума адаптивным фильтром Винера несколько снизила погрешность σ_{rel} последующей реконструкции смазанного изображения методом регуляризации Тихонова, но практически не повлияла на погрешность реконструкции методом параметрической фильтрации Винера.

Было рассмотрено также зашумление изображения импульсным шумом (типа “соль и перец”) и предварительная фильтрация его медианной фильтрацией (функция medfilt2.m) [21]. В этом случае роль предварительной фильтрации оказалась значительней. Это говорит о том, что методы регуляризации Тихонова и параметрической фильтрации Винера сами фильтруют в определенной степени гауссовый шум, но довольно слабо фильтруют импульсный шум (его лучше фильтрует медианная, а также ранговая фильтрация). Эти выводы подтвердились обработкой портретного изображения cameraman.tif.

Обработка цветных изображений. Обработка цветных изображений в системе MatLab обычно осуществляется способом покомпонентной обработки (RGB-изображение разбивается на R, G и B компоненты, а после обработки снова соединяется в RGB-изображение) [2–4, 9, 10, 15] или способом векторной обработки (изображение обрабатывается как трехмерная матрица) [9, 10]. Второй способ особенно удобен при программировании в системе MatLab.

Разработаны m-функции smearing.m, desmearingf.m, desmearingq.m и rmsd.m для моделирования смазывания изображения с усечением, размытием и поворотом, для реконструкции смазанного изображения методом ПФ и квадратур с регуляризацией Тихонова и для вычисления σ_{rel} применительно как к серому, так и цветному (RGB) изображению, причем для обработки RGB-изображения использован способ векторной обработки. Разработана также m-функция normnoise.m для зашумления как серых, так и цветных изображений гауссовым (нормальным) шумом в противовес имеющейся в пакете IPT MatLab m-функции imnoise.m, создающей некоторые неудобства при обращении к ней [10, с. 156]. Выполнена обработка с помощью методики “усечение–размытие–поворот” RGB-изображений pears.png и football.jpg, которая подтвердила выводы, сделанные выше по отношению к gray-изображениям textp.bmp и cameraman.tif.

Было также выполнено восстановление реальных смазанных изображений, а именно, сфотографированного со сдвигом фотоаппарата серого текста (типа textp.bmp на рис. 1а) и цветного изображения pair.jpg двух движущихся персон, показавшее, что качество полученных восстановленных изображений выше, чем исходных смазанных изображений.

Обработка дефокусированного изображения

Прямая и обратная задачи дефокусирования изображения в случае пространственно-инвариантной (разностной) ФРТ $h(x, y)$ описываются соотношением (2). В нем $h(x, y)$ в простейшем случае, когда ФРТ есть однородный круг постоянного радиуса ρ, записывается в виде [2, 3, 6–8, 11, 15]

$$h(x,y)=\begin{cases}1/\pi\rho^2, & \sqrt{x^2+y^2}\le\rho,\\ 0, & \sqrt{x^2+y^2}>\rho.\end{cases} \tag{10}$$

В более реальном случае ФРТ имеет вид гауссианы [11, 13]

$$h(x,y)=\frac{1}{2\pi\sigma^2}e^{-\left(x^2+y^2\right)/2\sigma^2}. \tag{11}$$

Однако ФРТ может иметь еще более общий вид [12, 16–18].

Прямая задача. Рассмотрим прямую задачу (моделирование дефокусирования изображения) в дискретном виде. Пусть нам дана матрица исходного неискаженного изображения $w_{m\times n}$ и необходимо смоделировать матрицу дефокусированного изображения g. Пусть при этом ФРТ h – любого вида, но пространственно-инвариантная и заключенная в пределах квадратного окна $2r \times 2r$, а вне окна равная нулю, т. е. h описывается матрицей $(2r + 1)\times(2r + 1)$. Если формировать дефокусированное изображение g такого же размера, как и исходное изображение w, то соотношение (3) более подробно запишется в виде (при замене интеграла в (8) конечной суммой по пикселам)

$$g(j,i)=\frac{1}{Q}\sum_{j'=j-r}^{j+r}\sum_{i'=i-r}^{i+r}h(i-i',j-j')w(j',i'),\quad j=1,2,\ldots,m,\quad i=1,2,\ldots,n. \tag{12}$$

В выражении (12) для удобства введен нормирующий делитель

$$Q=\sum_{i=-r}^{r}\sum_{j=-r}^{r}h(i,j), \tag{13}$$

что позволяет упростить записи формул (10), (11)

$$h(i,j)=\begin{cases}1, & \sqrt{i^2+j^2}\le r,\\ 0, & \sqrt{i^2+j^2}>r.\end{cases} \tag{10$'$}$$

$$h(i,j)=e^{-\left(i^2+j^2\right)/2\sigma^2}. \tag{11$'$}$$

Заметим, что в выражении (12) использовано "координатное соглашение" [10, с. 29, рис. 2.1(б)], согласно которому у матриц g и w первый индекс – это номер строки, а второй – номер столбца, причем начальные индексы равны единице. Это правило не касается h, поскольку h – это функция.

Проанализируем выражение (12). Видим, что при $i = n - r + 1, \ldots, n$ (вблизи правой границы изображений w и g) требуется знание значений $w(j, n - 2r + 1), \ldots, w(j, n + r)$, однако значения $w(j, n + 1), \ldots, w(j, n + r)$ (за правой границей w) неизвестны. Аналогично требуются, но неизвестны, значения w левее, выше и ниже границы изображения w на величину r пикселов. Чтобы преодолеть дефицит информации, в работах [9, 10, 17–20], как уже было отмечено, предложены и используются так называемые "граничные условия": от простейшего *zero* до *anti-reflective*, в которых делаются различные предположения относительно недостающих значений w. Заметим, что такой эффект "недостачи" имеет место только при обработке модельных изображений, а при получении реальных искаженных изображений сама природа поставляет необходимые значения w в искаженное изображение g.

Усечение дефокусированного изображения. Предлагаем иначе решить вопрос, связанный с недостатком значений w, а именно, вместо соотношения (12) использовать соотношение (4)

$$\hat{g}(j,i)=\frac{1}{Q}\sum_{j'=j-r}^{j+r}\sum_{i'=i-r}^{i+r}h(i-i',j-j')w(j',i'),\quad j=1+r,2+r,\ldots,m-r,\quad i=1+r,2+r,\ldots,n-r. \tag{14}$$

Отличие соотношения (14) от (12) состоит в том, что согласно (14) моделируется изображение $\hat{g}$ меньшего размера, чем w. При этом все необходимые значения w известны и не нужно прибегать к такому приему, как "граничные условия". Правда, изображение $\hat{g}$ получается несколько меньше, чем w (на r пикселов сверху, снизу, слева и справа). Но, во-первых, эффект уменьшения практически невелик (обычно $m \approx n \approx 300 \div 1000$, а $r \sim 10$), а, во-вторых, при решении обратной задачи, например, методом регуляризации Тихонова в качестве искомого решения w_α используется нормальное псевдорешение [3, 8], которое существует и является единственным, несмотря на то, что задача отыскания w_α является недоопределенной (область изображения $\hat{g}$ у́же области изображения w).

Заметим, что выражение (14) не вполне удобно для программирования в системе MatLab, поскольку в ней начальные индексы у массивов должны равняться единице. Более удобным является выражение

$$\hat{g}(j,i)=\frac{1}{Q}\sum_{j'=j}^{j+2r}\sum_{i'=i}^{i+2r}h(i-i'+r,j-j'+r)w(j',i'),\quad j=1,2,\ldots,m-2r,\quad i=1,2,\ldots,n-2r. \tag{15}$$

Однако и выражение (15) недостаточно удобно для программирования в системе MatLab, поскольку в нем фигурируют два массива (матрицы) $\hat{g}$ и w и функция h, а MatLab наиболее эффективен, когда оперирует лишь с матрицами. Поэтому введем матрицу M, связанную с функцией h следующим образом:

$$M(j,i)=h(i-r-1,j-r-1),\quad j=1,2,\ldots,2r+1,\quad i=1,2,\ldots,2r+1. \tag{16}$$

Тогда выражение (15) запишется в виде [6]

$$\hat{g}(j,i)=\frac{1}{Q}\sum_{j'=j}^{j+2r}\sum_{i'=i}^{i+2r}M(j-j'+2r+1,i-i'+2r+1)w(j',i'),\quad j=1,2,\ldots,m-2r,\ i=1,2,\ldots,n-2r. \tag{17}$$

Назовем моделирование дефокусированного изображения $\hat{g}$ на основе соотношений (14)–(17) приемом усечения дефокусированного изображения.

Размытие краев дефокусированного изображения. Для снижения эффекта ложных волн (эффекта Гиббса) на восстанавливаемом изображении, обусловленного резким падением до нуля интенсивностей на краях изображения $\hat{g}$, введем размытие краев у дефокусированного изображения (ср. (7) и (17)) [6]

$$g(j,i)=\frac{1}{Q}\sum_{j'=j}^{j+2r}\sum_{i'=i}^{i+2r}M(j-j'+2r+1,i-i'+2r+1)u(j',i'),$$
$$j=1,2,\ldots,m+2r,\; i=1,2,\ldots,n+2r, \quad (18)$$

где u – матрица $(m+4r)\times(n+4r)$, введенная для удобства расчетов путем добавления к матрице w по $2r$ нулей слева, справа, сверху и снизу (это можно сделать с помощью m-функции padarray.m).

Для моделирования дефокусированных серого или цветного изображений с усечением согласно (17) или с размытием краев согласно (18) разработана в рамках системы MatLab m-функция defocusing.m.

Обратная задача. Под обратной задачей подразумевается задача восстановления изображения w по дефокусированному изображению g и известной (некоторым образом) функции рассеяния точки h.

Если ФРТ пространственно-инвариантная (разностная): $h = h(x-\xi, y-\eta)$, то задача восстановления сводится к решению 2-мерного ИУ Фредгольма I рода типа свертки (2) (некорректная задача). Методы решения уравнения (2) – это методы инверсной фильтрации, псевдоинверсной фильтрации, оптимальной и параметрической фильтрации Винера, регуляризации Тихонова, алгоритмы Люси–Ричардсона, "слепой" деконволюции и др. [1–3, 7–18]. Эти методы используют в той или иной мере 2-мерное ПФ.

В методе параметрической фильтрации Винера [9, с. 392], [10, с. 184] решение уравнения (2) равно

$$w_K(\xi,\eta)=\frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}W_K(\omega_1,\omega_2)e^{-i(\omega_1\xi+\omega_2\eta)}d\omega_1 d\omega_2, \quad (19)$$

где

$$W_K(\omega_1,\omega_2)=\frac{H^*(\omega_1,\omega_2)G(\omega_1,\omega_2)}{|H(\omega_1,\omega_2)|^2+K}, \quad (20)$$

$$H(\omega_1,\omega_2)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}h(x,y)e^{i(\omega_1 x+\omega_2 y)}dxdy, \quad (21)$$

$$G(\omega_1,\omega_2)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}g(x,y)e^{i(\omega_1 x+\omega_2 y)}dxdy, \quad (22)$$

$K \ge 0$ – некоторая константа (параметр, дающий приближение отношения энергетических спектров шума и неискаженного изображения).

В методе регуляризации Тихонова [3, 4, 7–9, 11, 14, 15] решение равно

$$w_\alpha(\xi,\eta)=\frac{1}{4\pi^2}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}W_\alpha(\omega_1,\omega_2)e^{-i(\omega_1\xi+\omega_2\eta)}d\omega_1 d\omega_2, \quad (23)$$

где

$$W_\alpha(\omega_1,\omega_2)=\frac{H^*(\omega_1,\omega_2)G(\omega_1,\omega_2)}{|H(\omega_1,\omega_2)|^2+\alpha(\omega_1^2+\omega_2^2)^p}, \quad (24)$$

$\alpha > 0$ – параметр регуляризации, $p \ge 0$ – порядок регуляризации.

Метод параметрической фильтрации Винера реализован в системе MatLab7 в виде m-функции deconvwnr.m [9, с. 397], [10, с. 188]. Авторами статьи разработана собственная m-функция refocusing.m, в которой реализован метод регуляризации Тихонова (выражения (21)–(24) в дискретном виде) применительно к дефокусированному (серому или цветному) изображению g с усечением или с размытием.

Численные примеры. Было промоделировано дефокусирование (с помощью m-функции defocusing.m) текстового серого изображения textp.bmp, представленного на рис. 1а. Использовалась ФРТ в виде однородного диска радиуса $r = 10$ (см. (10′)). На рис. 2а представлено дефокусированное изображение 598×670 с усечением согласно (17), а на рис. 2б – дефокусированное изображение 638×710 с размытием краев согласно (18). На рис. 2в представлена реконструкция изображения 2а методом 2-мерного ПФ с регуляризацией Тихонова ($\alpha = 10^{-2}$, $p = 2$, $\sigma_{\text{rel}} = 0{,}4915$), а на рис. 2г – реконструкция изображения 2б тем же методом ($\alpha = 10^{-7}$, $p = 2$, $\sigma_{\text{rel}} = 0{,}0132$). Для сравнения на рис. 2д приведено изображение, дефокусированное с помощью m-функций fspecial('disk'...) и imfilter(...'symmetric'), а на рис. 2е – реконструкция изображения 2д параметрической фильтрацией Винера ($K = 10^{-5}$, $\sigma_{\text{rel}} = 0{,}0906$).

Рис. 2в показывает, что реконструкция с усечением выполнена с большой погрешностью (с темным фоном и ложными волнами), а реконструкция с размытием краев (рис. 2г) – практически без погрешностей. Что касается параметрической фильтрации Винера, то она реконструировала изображение (рис. 2е) с ложными волнами и с погрешностью σ, заметно большей, чем у изображения 2г. Это говорит о том, что использование приема “размытие краев изображения” может заметно повысить точность реконструкции. Кроме того, эти результаты указывают на то, что регуляризация Тихонова в принципе точнее параметрической фильтрации Винера (ср. (20) и (24)).

Было также обработано (дефокусировано и затем рефокусировано) цветное изображение flower.bmp и сделанные выводы подтвердились.

Была выполнена обработка серого изображения textp.bmp и RGB-изображения flower.bmp при добавлении к дефокусированию гауссова, а также импульсного шума. Результаты такой обработки аналогичны обработке смазанного зашумленного изображения: при 1%-ном и большем шуме методы регуляризации Тихонова (с размытием) и параметрической фильтрации Винера дают примерно одинаковую погрешность σ; дополнительная фильтрация шума адаптивным (оконным) фильтром Винера, если шум гауссов, дает небольшое понижение погрешности σ и заметное ее понижение при использовании медианной фильтрации, если шум импульсный. Это подтверждает вывод о том, что методы регуляризации Тихонова (с размытием) и параметрической фильтрации Винера, устраняя дефокусирование, “попутно” в определенной степени сами фильтруют гауссовый шум, но довольно слабо фильтруют импульсный шум (в этом случае целесообразно добавить медианную фильтрацию).

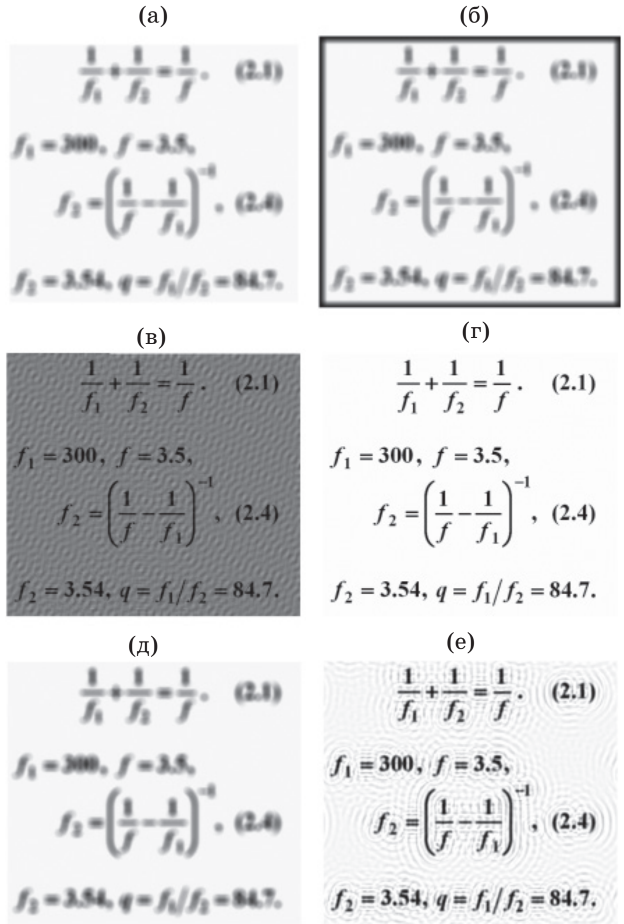

Рис. 2. Прямая и обратная задачи обработки дефокусированного серого текстового изображения textp.bmp ($r = 10$ пикселов, без шума). а – дефокусированное изображение с усечением g_1 598×670 согласно (17). б – дефокусированное изображение с размытием краев g_2 638×710 согласно (18). в – реконструкция усеченного изображения g_1 методом 2-мерного ПФ и регуляризации Тихонова. г – реконструкция изображения g_2 с размытием краев методом 2-мерного ПФ и регуляризации Тихонова. д – дефокусированное изображение defoc 618×690 с использованием функций fspecial.m и imfilter.m. е – реконструкция изображения defoc методом параметрической фильтрации Винера.

О сравнительных возможностях пакета PhotoShop. Для обработки изображений часто используется пакет Adobe PhotoShop, а также Paint Shop Pro, PhotoImpact и др. [22]. Дадим краткое сравнение возможностей этих пакетов и пакета IPT MatLab, в том числе, методики, изложенной в данной работе. Пакеты PhotoShop и др. позволяют хорошо и удобно менять яркость, контраст, оттенки, выполнять кадрирование, ретушь, формировать рамку, устранять царапины, а также (по тематике данной работы) моделировать размытие в движении, размытие по Гауссу и добавлять шум. Однако формирование размытия и добавление шума – это прямые задачи. Что же касается обратных (наиболее сложных) задач, то пакеты PhotoShop и др. позволяют фильтровать шум (медианным фильтром), но не устраняют размытие (смазывание по терминологии данной работы) и довольно слабо устраняют дефокусирование (фильтром “Контурная резкость”, повышающим контраст смежных пикселов, т. е. усиливающим резкость границ на изображении).

Это можно объяснить тем, что в пакете IPT MatLab и в методике данной работы для решения обратных задач привлечен мощный математический аппарат (решение некорректных 1-мерных и 2-мерных интегральных уравнений Фредгольма I рода методами регуляризации Тихонова, фильтрации Винера и др.), а в пакете PhotoShop и др. использованы лишь такие довольно простые математические методы, как медианная фильтрация. Кстати, если искажение не есть смазывание или дефокусирование в чистом виде, т.е. ФРТ имеет произвольную форму [16, 17], то MatLab позволяет решить и такую обратную задачу, в противоположность пакету PhotoShop и др.

О быстром алгоритме реконструкции смазанных изображений. Важным является вопрос о реализуемости предложенной в работе методики в реальном масштабе времени. Рассмотрим кратко его на примере быстродвижущейся под углом $\theta = 0$ цели (например, самолета).

Реконструкцию изображения методом квадратур с регуляризацией Тихонова (вариант 4) [1] можно записать в виде $w_\alpha = B_\alpha g$, где

$$B_\alpha = \left(\alpha E + A^T A\right)^{-1} A^T$$

– матрица $n\times(n + \Delta)$, которая может быть рассчитана заранее, и реконструкция изображения сведется к умножению "заготовленной" матрицы B_α на каждый из m векторов-строк смазанного изображения g.

Оценим количество операций и требуемое время. Умножение матрицы B_α размера $n\times(n + \Delta)$ на вектор g длиной $n + \Delta$ запишем в виде

$$(w_\alpha)_i = \sum_{k=1}^{n+\Delta} (B_\alpha)_{ik}\, g_k, \quad i = 1, 2, \ldots, n.$$

На это потребуется $n(n + \Delta)$ умножений (и сложений). А на выполнение таких операций для всех m строк изображения потребуется $mn(n + \Delta)$ умножений. Например, при $m = n = 400$ и $\Delta \ll n$ потребуется $\approx$ 64 млн умножений. Если скорость компьютера ~ 1 млрд оп/с, то реконструкция изображения займет ~ 0,1 с.

Замечание. Матрица B_α не зависит от номера строки j (при $\theta = 0$), но зависит от величины смаза Δ и параметра регуляризации α, а также от значений m и n. Размер $m\times n$ обычно фиксирован, а величины Δ и α заранее неивестны. Поэтому следует заранее заготовить несколько матриц B_α для ряда типичных значений Δ и α.

Если речь идет, например, о самолете – нарушителе границы, то служба первичного обнаружения цели должна оценить скорость, направление движения и высоту самолета и по этим данным служба обработки изображения оценит Δ (при известной экспозиции) и вызовет матрицу B_α с некоторым типичным значением α (или две–три таких матрицы). И тогда за долю секунды, т. е. в реальном масштабе времени будет выполнена реконструкция изображения самолета. Такая обработка была выполнена (файл изображения plane.bmp) и она подтвердила вышесказанное.

Более подробно затронутые в данной работе вопросы будут изложены в книге: Сизиков В.С. "Обратные прикладные задачи и MatLab", подготовленной к печати в изд-ве "Лань".

Работа выполнена при поддержке РФФИ (грант 09-08-00034а).

ЛИТЕРАТУРА

1. *Сизиков В.С., Римских М.В., Мирджамолов Р.К.* Реконструкция смазанных и зашумленных изображений без использования граничных условий // Оптический журнал. 2009. Т. 76. № 5. С. 38–46.
2. *Сизиков В.С., Белов И.А.* Реконструкция смазанных и дефокусированных изображений методом регуляризации // Оптический журнал. 2000. Т. 67. № 4. С. 60–63.
3. *Сизиков В.С.* Математические методы обработки результатов измерений. СПб.: Политехника, 2001. 240 с.
4. *Petrov Yu.P., Sizikov V.S.* Well-Posed, Ill-posed, and Intermediate Problems with Applications. Leiden–Boston: VSP, 2005. 234 p.
5. *Сизиков В.С.* Применение аппарата интегральных уравнений для реконструкции искаженных изображений // Сб. тез. конф. "Интегральные уравнения – 2009". Киев: Изд-во ИПМЭ, 2009. С. 128–130.
6. *Сизиков В.С.* Интегральные уравнения и новый прием "усечение–размытие–поворот" в реконструкции искаженных изображений // Proc. Intern. Conf. "Integral Equations – 2010". Lviv: PAIS, 2010. P. 138–142.
7. *Тихонов А.Н., Гончарский А.В., Степанов В.В.* Обратные задачи обработки фотоизображений // Некоторые задачи естествознания / Под ред. Тихонова А.Н., Гончарского А.В. М.: Изд-во МГУ, 1987. С. 185–195.
8. *Бакушинский А.Б., Гончарский А.В.* Некорректные задачи. Численные методы и приложения. М.: Изд-во МГУ, 1989. 199 с.
9. *Гонсалес Р., Вудс Р.* Цифровая обработка изображений. М.: Техносфера, 2006. 1072 с.

10. *Гонсалес Р., Вудс Р., Эддинс С.* Цифровая обработка изображений в среде MATLAB. М.: Техносфера, 2006. 616 с.

11. *Арефьева М.В., Сысоев А.Ф.* Быстрые регуляризирующие алгоритмы цифрового восстановления изображений // Вычислит. методы и программирование. 1983. В. 39. С. 40–55.

12. *Горшков А.В.* Улучшение разрешения изображений при обработке данных физического эксперимента и нахождение неизвестной аппаратной функции по программам пакета Reimage // Приборы и техника эксперимента. 1995. № 2. С. 68–78.

13. *Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А.* Цифровая обработка изображений в информационных системах. Новосибирск: Изд-во НГТУ, 2000. 168 с.

14. *Воскобойников Ю.Е., Литасов В.А.* Устойчивый алгоритм восстановления изображения при неточно заданной аппаратной функции // Автометрия. 2006. Т. 42. № 6. С. 3–15.

15. *Ягола А.Г., Кошев Н.А.* Восстановление смазанных и дефокусированных цветных изображений // Вычислит. методы и программирование. 2008. Т. 9. С. 207–212.

16. *Donatelli M., Estatico C., Martinelli A., Serra-Capizzano S.* Improved image deblurring with anti-reflective boundary conditions and re-blurring // Inverse problems. 2006. V. 22. P. 2035–2053.

17. *Wendykier P., Nagy J.G.* Image Processing on Modern CPUs and GPUs. Technical Report TR-2008-023, http://www.mathcs.emory.edu/technical-reports/techrep-00148.pdf

18. *Palmer K., Nagy J., Perrone L.* Iterative methods for image restoration: Matlab object oriented approach, 2002, http://citeseer.ist.psu.edu/lee02iterative.html

19. *Christiansen M., Hanke M.* Deblurring methods using antireflective boundary conditions, 2006, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.70.9837&rep=rep1&tipe=pdf

20. *Arico A., Donatelli M., Nagy J., Serra-Capizzano S.* The Anti-Reflective Transform and Regularization by Filtering. Technical report TR-2007-006-A, ftp://ftp.mathcs.emory.edu/pub/techreport/TR-2007-006-A.pdf

21. *Дьяконов В., Абраменкова И.* MATLAB. Обработка сигналов и изображений. СПб.: Питер, 2002. 608 с.

22. *Эндрюс Ф.* Цифровая фотография. М.: ЗАО “Росмэн-Издат”, 2005. 192 с.