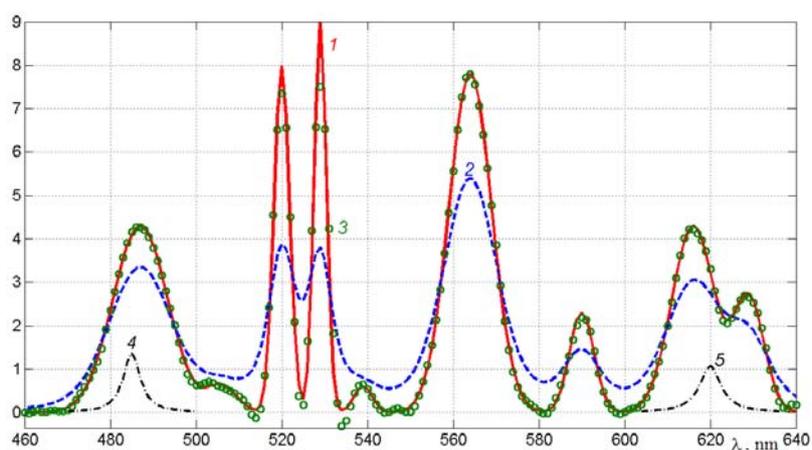


В.С. Сизиков

А.В. Лавров

## УСТОЙЧИВЫЕ МЕТОДЫ МАТЕМАТИКО-КОМПЬЮТЕРНОЙ ОБРАБОТКИ ИЗОБРАЖЕНИЙ И СПЕКТРОВ



Санкт-Петербург  
2018

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Университет ИТМО

**В.С. Сизиков**

**А.В. Лавров**

**УСТОЙЧИВЫЕ МЕТОДЫ  
МАТЕМАТИКО-КОМПЬЮТЕРНОЙ  
ОБРАБОТКИ ИЗОБРАЖЕНИЙ И СПЕКТРОВ**

**РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО**

**по направлениям подготовки 09.03.04 Программная инженерия, 09.04.01 Информатика  
и вычислительная техника, 09.04.02 Информационные системы и технологии  
в качестве учебного пособия для реализации основных образовательных программ  
высшего образования магистратуры**

 **УНИВЕРСИТЕТ ИТМО**

**Санкт-Петербург  
2018**

Сизиков В.С., Лавров А.В. Устойчивые методы математико-компьютерной обработки изображений и спектров. Учебное пособие. – Санкт-Петербург: Университет ИТМО. 2018 . – 70 с.

Рецензенты:

Сидоров Д. Н. – д.ф.-м.н., профессор ИрГУ, профессор РАН и вед. научный сотрудник ИСЭ РАН, г. Иркутск;

Манойлов В.В. – д.ф.-м.н., вед. научный сотрудник ИАП РАН, профессор Университета ИТМО, г. Санкт-Петербург.

В учебном пособии приведено описание методов, алгоритмов и MatLab-программ для восстановления искаженных (смазанных, дефокусированных, зашумленных) изображений, а также восстановления тонкой структуры спектров (сглаженных аппаратной функцией спектрометра или из-за перекрытия спектральных линий). Задачи восстановления изображений и спектров имеют общее: математический аппарат (интегральные уравнения, преобразование Фурье, квадратуры, регуляризация, фильтрация), а также разработка программ. Учебное пособие предназначено для бакалавров, магистрантов и аспирантов, изучающих дисциплины «Математическое моделирование и численные методы», «Методы обработки изображений и спектров», «Прикладная математика», «Программирование», а также для преподавателей и научных сотрудников.



**Университет ИТМО** – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2018  
© Сизиков В.С., Лавров А.В., 2018

## Содержание

1. Введение .....	4
2. Об изображениях .....	10
3. Обработка изображений .....	12
3.1. Моделирование смазывания изображений (прямая задача) .....	12
3.2. Восстановление смазанных изображений (обратная задача) .....	15
3.3. Моделирование дефокусирования изображений (прямая задача) .....	21
3.4. Восстановление дефокусированных изображений (обратная задача) ...	24
3.5. Шумы на изображениях .....	27
4. Обработка спектров .....	37
4.1. Введение в спектральный анализ .....	37
4.2. Восстановление непрерывных спектров .....	40
4.3. Разделение перекрывающихся спектральных линий .....	48
5. Приложение (некоторые MatLab-программы) .....	59
Список использованной литературы .....	64

## 1. Введение

В данном учебном пособии изложены прямые и обратные прикладные задачи восстановления искаженных изображений и спектров. Дано физическое, техническое, математическое и программное описание задач. Изложены некоторые устойчивые методы решения уравнений, в основном, интегральных. Приведены тексты ряда программ в системе MatLab и результаты решения прямых и обратных задач.

**Определение.** *Прямой задачей* называется моделирование (формирование) выходного процесса (измеренного изображения или спектра), полученного измерительным устройством (фотоаппаратом или спектрометром) по входному процессу (и аппаратной функции устройства, см. дальше).

**Замечание.** Если прямая задача рассчитывается на вычислительном устройстве (персональном компьютере), то под моделированием выходного процесса будем подразумевать его компьютерный расчет, а если прямая задача измеряется на реальном измерительном устройстве, то под формированием выходного процесса будем подразумевать его получение непосредственно измерительным устройством.

### Примеры прямых прикладных задач [1–3].

1. Пусть дан некоторый, например, временной процесс  $y(t)$ . В качестве первого примера прямой задачи рассмотрим вычисление преобразования Фурье, или *спектра процесса*:

$$Y(\omega) = \int_{-\infty}^{\infty} y(t)e^{i\omega t} dt, \quad -\infty < \omega < \infty, \quad (1)$$

где  $\omega$  – круговая частота Фурье.

На рис. 1 – пример исходного процесса  $y(t)$  и его преобразования Фурье в виде модуля  $|Y(\omega)|$ . Данный пример в системе MatLab записан в виде файла:

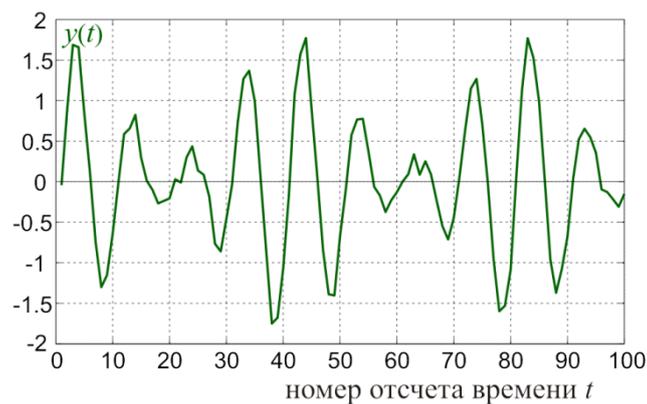
```
%Fig1.m, Сизиков, головная программа, MatLab 7 или 8.
%Временной процесс в виде трех синусоидальных гармоник и шума.
clc, clear %очистка Command Window и Workspace
t = 0:0.0005:1; %2000 шагов по времени t
y = 0.4*sin(2*pi*150*t) + sin(2*pi*200*t) + 0.4*sin(2*pi*250*t);
randn('state',0) %инициализация генератора случайных чисел
y1 = y + 0.1*randn(size(t)); %добавление нормального шума [0,0.05]
%График процесса (первые 100 отсчетов):
figure %формирование нового графического окна
plot(y1(1:100), 'Color', [0,0.5,0], 'LineWidth', 2.5) %график y1 (зел.)
grid on %сетка
hold on %продолжение графика (с другими параметрами)
```

```

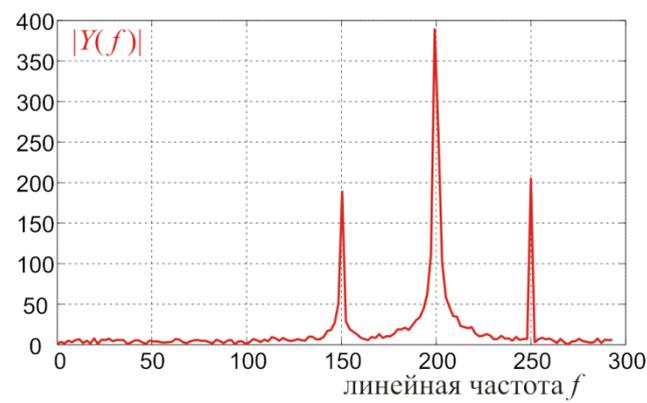
plot(zeros(1,100),'-k','LineWidth',1.2) %линия нулевого уровня (черн.)
text(2.25,1.87,'\ityy({\itt})','FontSize',22,'FontName',...
'Times New Roman','Color',[0,0.5,0]); %текст на графике (зел.)
text(65,-2.3,'номер отсчета времени {\itt}','FontSize',18,...
'FontName','Times New Roman');
axis([0,100,-2,2]); %диапазоны координат по осям
hold off %отмена продолжения построения графика
%Спектр (ПФ) процесса:
Y1 = fft(y1,1024); %БПФ по первым 1024 отсчетам
M = abs(Y1); %модуль ПФ (АЧХ спектра) 1x1024 double
f = (0:150)*2000/1024; %первое 151 значение линейной частоты
figure
plot(f,M(1:151),'-r','LineWidth',2.5) %график |Y1| (красн.)
grid on
text(10,380,'|{\itY}({\it f})|','FontSize',24,...
'FontName','Times New Roman','Color','r');
text(220,-30,'линейная частота {\it f}','FontSize',18,...
'FontName','Times New Roman');
axis([0,300,0,400]);

```

Отметим, что текст программы Fig1.m специально содержит подробности, чтобы помочь тем, кто еще недостаточно освоил MatLab.



а



б

Рис. 1. а – исходный зашумленный вещественный процесс  $y(t)$ ;  
б – модуль комплексного преобразования Фурье  $|Y(f)|$

Из рис. 1б мы видим, что в спектре четко отобразились три гармоники, хотя по рис. 1а трудно судить о их наличии.

2. Моделирование смазывания изображения по формуле:

$$g(x, y) = \frac{1}{\Delta} \int_x^{x+\Delta} w(\xi, y) d\xi, \quad (2)$$

где  $w(\xi, y)$  – заданное распределение интенсивности по истинному (неискаженному) изображению,  $g(x, y)$  – распределение интенсивности по смазанному изображению, при этом ось  $x$  (а также  $\xi$ ) направлена вдоль смаза, а ось  $y$  – перпендикулярно смазу,  $\Delta$  – величина смаза (подробности далее). Если же смазывание изображения произошло «естественным» путем (в результате сдвига фотоаппарата, движения объекта и т.д.), то функция  $w(\xi, y)$  будет неизвестна (хотя потенциально существует), а функция  $g(x, y)$  формируется (но не рассчитывается) устройством обработки изображения (фотоаппаратом и т.д.).

На рис. 2 в качестве примера представлено неискаженное томографическое изображение (фантом), а также смазанное (и зашумленное) изображение.

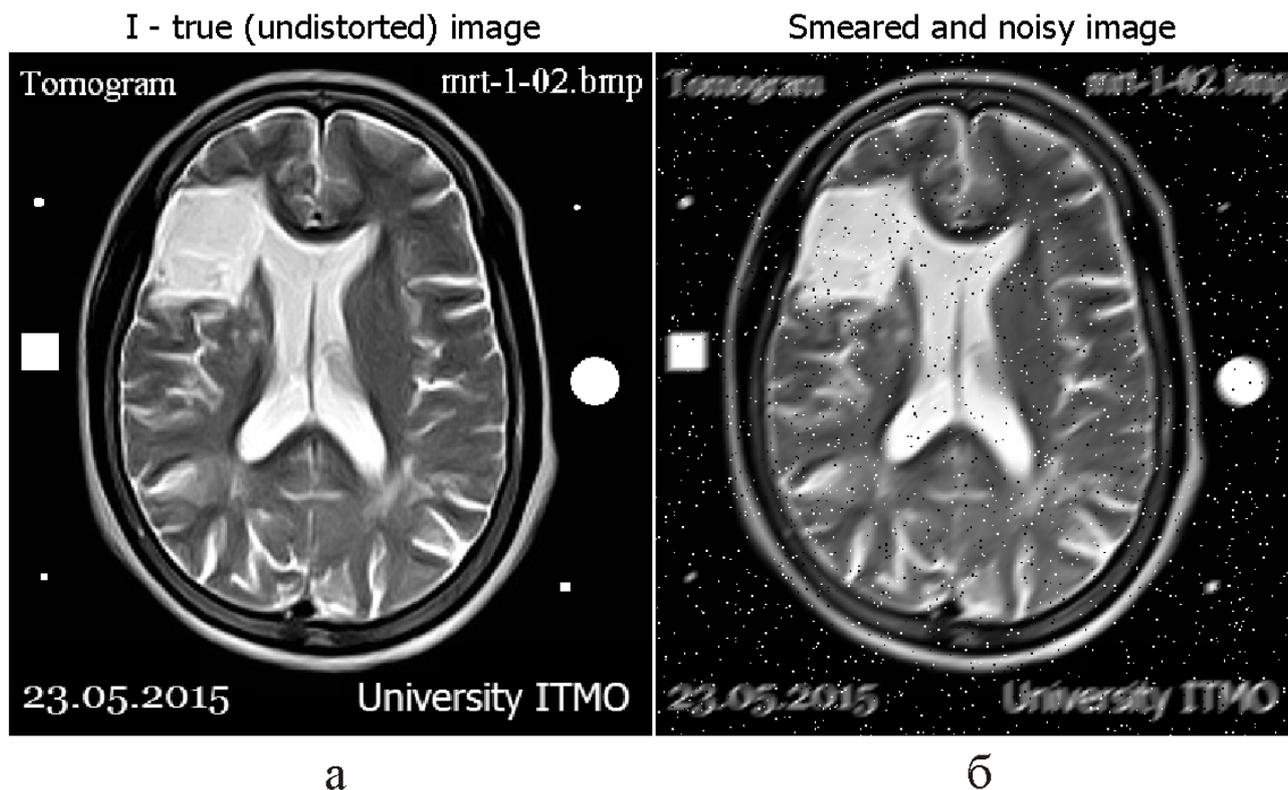


Рис. 2. а – неискаженная томограмма; б – смазанная и зашумленная томограмма

Этот пример рассчитан по следующей MatLab-программе:

```
%Fig2.m, Сизиков, головная программа, MatLab.
%Пример моделирования смаза и зашумления изображения (прямая задача).
clc,clear %очистка Command Window и Workspace
%Чтение томографического изображения (томограммы):
RGB = imread('mrt-1-02.bmp'); %407x380x3 uint8
```

```

I = rgb2gray( RGB ); %преобразование в полутоновое (серое) изображение
%Вывод изображения (с повышенным контрастом) в 1-е подокно:
figure
subplot(1,2,1), imshow(I,[])
title('I - true (undistorted) image','FontSize',18,...
      'FontName','Tahoma')
%Создание маски (ФПТ, PSF) для смазывания изображения:
PSF = fspecial('motion',7,35); %смаз длиной 7 пкс под углом 35 град.
%Смазывание изображения:
S = imfilter(I,PSF,'circular'); %использование опции 'circular'
%Зашумление смазанного изображения импульсным шумом ("соль и перец"):
SN = imnoise(S,'salt & pepper',0.02); %зашумлено 2% изображения
%Вывод смазанного и зашумленного изображения во 2-е подокно:
subplot(1,2,2), imshow(SN,[])
title('Smearred image','FontSize',18,'FontName','Tahoma')

```

### 3. Моделирование измерения непрерывного спектра спектрометром:

$$u(\lambda) = \int_a^b K(\lambda, \lambda') z(\lambda') d\lambda', \quad c \leq \lambda \leq d, \quad (3)$$

где  $z(\lambda)$  – заданное истинное распределение интенсивности по спектру в функции длины волны  $\lambda$ ;  $K(\lambda, \lambda')$  – аппаратная функция (АФ) спектрометра;  $u(\lambda)$  – рассчитанное распределение интенсивности по измеренному спектру (если же спектр  $u(\lambda)$  получен в результате физического эксперимента, то он регистрируется непосредственно спектрометром, но не рассчитывается). Подробности данной задачи спектроскопии см. в п. 4.2.

На рис. 3 в качестве примера дано распределение энергии по истинному спектру  $z(\lambda)$ , сглаженное измеренное спектрометром распределение энергии  $u(\lambda)$  и два сечения неразностной аппаратной функции  $K(485, \lambda')$  и  $K(620, \lambda')$ .

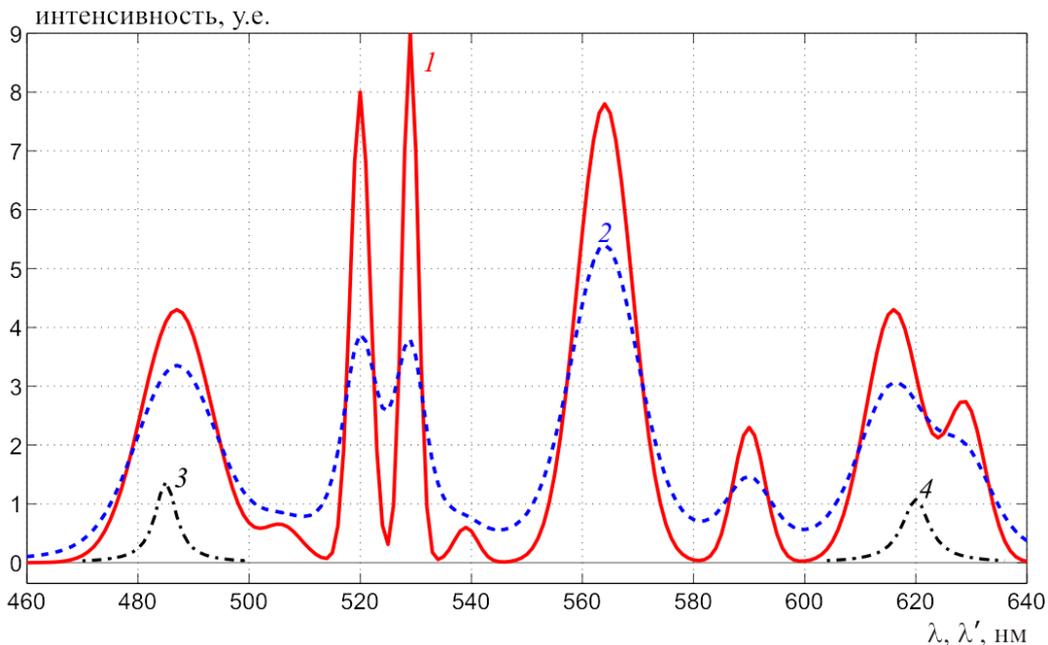


Рис. 3. 1 – истинный спектр  $z(\lambda)$ ; 2 – измеренный спектр  $u(\lambda)$ ;  
3 –  $10K(485, \lambda')$ ; 4 –  $10K(620, \lambda')$

В приведенных примерах  $y(t)$ ,  $w(\xi, y)$  и  $z(\lambda)$  – входные процессы, а  $Y(\omega)$ ,  $g(x, y)$  и  $u(\lambda)$  – выходные процессы, рассчитанные путем вычисления интегралов (1)–(3) или полученные в результате измерений.

Перейдем к обратным (более сложным, но и более важным) задачам.

**Определение.** *Обратной задачей* называется восстановление входного процесса по измеренному выходному процессу (и аппаратной функции).

**Примеры обратных прикладных задач.** Это те же примеры 1–3, но записанные в иной форме.

1. *Восстановление исходного процесса*  $y(t)$  по известному спектру  $Y(\omega)$  на основе соотношения (1), записанного в виде:

$$\int_{-\infty}^{\infty} y(t) e^{i\omega t} dt = Y(\omega), \quad -\infty < \omega < \infty. \quad (4)$$

Казалось бы, мы лишь поменяли местами  $Y(\omega)$  и интеграл, т.е. левую и правую части в (1), но мы получили в (4) существенно новый результат. Соотношение (4) можно рассматривать как интегральное уравнение относительно  $y(t)$ , при этом  $Y(\omega)$  – правая часть, а  $e^{i\omega t}$  – ядро уравнения. Отметим, что вычислять интеграл в (1) обычно значительно проще, чем решать интегральное уравнение (4) (подробнее об интегральных уравнениях см. далее). Правда, уравнение (4) есть тот редкий случай, когда его решение находится аналитически (в виде обратного преобразования Фурье):

$$y(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} Y(\omega) e^{-i\omega t} d\omega, \quad -\infty < t < \infty. \quad (5)$$

2. *Восстановление распределения интенсивности* по истинному (неискаженному) изображению на основе измеренного смазанного изображения с распределением интенсивности  $g(x, y)$ . Задача сводится к решению одномерного интегрального уравнения Вольтерра I рода:

$$\frac{1}{\Delta} \int_x^{x+\Delta} w_y(\xi) d\xi = g_y(x) \quad (6)$$

при каждом значении  $y$ , играющем роль параметра, т.е. к решению набора одномерных интегральных уравнений (подробности – в п. 3.2).

3. *Восстановление истинного спектра*  $z(\lambda)$  на основе измеренного спектра  $z(\lambda)$  и аппаратной функции спектрометра  $K(\lambda, \lambda')$  – обратная задача спектроскопии. Задача заключается в решении интегрального уравнения Фредгольма I рода:

$$\int_a^b K(\lambda, \lambda') z(\lambda') d\lambda' = u(\lambda), \quad c \leq \lambda \leq d. \quad (7)$$

Решение уравнения (7) позволит разрешить близкие линии в спектре, выделить слабые линии, восстановить тонкую структуру спектра, другими слова-

ми, повысить разрешающую способность спектрометра математическим путем – это один из вариантов редуцированной проблемы Рэля (см. п. 4.2).

Обратные задачи формулируются также в виде *операторного уравнения*

$$Ay = f, \quad (8)$$

где  $A$  – некоторый оператор,  $y$  – искомая функция (входной процесс),  $f$  – правая часть (выходной процесс).

Решение уравнений (6)–(8) и др. позволяет, в принципе, математико-компьютерным путем восстановить входной процесс и тем самым повысить разрешающую способность измерительного устройства (фотоаппарата, телескопа, микроскопа, спектрометра, томографа, антенны и т.д.).

Сопряжение измерительного устройства с вычислительным устройством (компьютером, цифровым сигнальным процессором и т.д.) равносильно созданию нового, более совершенного измерительного устройства.

Яркий пример тому – восстановление дефокусированных изображений космических объектов, полученных на американском космическом телескопе «Хаббл» (КТХ) (см. рис. 4).



Рис. 4. Необработанное изображение галактики М100 (слева) и исправленное изображение (справа)

При этом сделаем следующее замечание. Обратные задачи можно разделить на *два типа* (ср. [1–3]).

Обратные задачи *первого типа* – это те задачи, в которых искомую функцию можно получить без математической обработки измерений (за счет совершенствования измерительного устройства). Примеры задач первого типа – это вышеизложенные примеры 2 и 3 (восстановление изображений и спектров).

Обратные задачи *второго типа* – это такие задачи, при решении которых никакое совершенствование измерительного устройства не позволит восстановить искомую функцию без математико-компьютерной обработки измерений. Пример задач второго типа – это непрямые методы ЯМР-томографии, рентгеновской компьютерной томографии (РКТ) и ИК-томографии [1–3]. Роль математических методов при решении задач второго типа гораздо важнее, чем при решении задач первого типа.

## 2. Об изображениях

Приведем основные понятия, связанные с изображениями и их обработкой (объекты, приборы, искажения, классы чисел) [1–7].

**Снимаемые объекты:** люди, животные, тексты, космические объекты (планеты, галактики), наблюдаемые в телескоп, земная поверхность, сканируемая со спутника, вирусы, наблюдаемые через микроскоп, движущиеся объекты (самолет, автомобиль, танк), номерной знак автомобиля и т.д. В результате съемки получаем *изображения объектов*.

**Приборы регистрации изображений:** цифровые фотоаппараты, видеокамеры, спутниковые и авиационные бортовые датчики, приборы медицинской визуализации, телескопы, микроскопы, томографы и т.д. Отметим также вычислительные устройства: персональные компьютеры (ПК), малогабаритные специализированные вычислительные устройства (СВУ).

Обычно изображение в цифровых устройствах регистрируется матрицей сенсоров на основе ПЗС-матрицы. *Сенсор* – это элемент, преобразующий освещенность в напряжение [5, с. 90], а *ПЗС* – полупроводниковый прибор с зарядовой связью, в котором осуществляется перенос заряда от элемента к элементу [5, с. 96], [7, с. 33].

При этом выполняются *оцифровка, дискретизация и квантование* изображения [5, 7]. Устройством, выполняющим эти операции, может являться цифровая фото- или видеокамера. В результате изображение будет представлено *матрицей интенсивностей* из  $t$  строк и  $n$  столбцов или  $t \times n$  пикселей (будем писать кратко: пкс). *Пиксель* – это точка на дискретном изображении (элемент изображения).

**Искажения изображений.** Будем рассматривать искажения, требующие сложной математической обработки (согласно [4]): смазывание, дефокусирование (расфокусировка), зашумленность изображения.

Причиной смазывания изображения объекта могут быть: смещение УРИ – устройства регистрации изображений (фотоаппарата, видеокамеры и т.д.), несогласование движения конвейера с деталями и камеры параллельного слежения, отличие вращения телескопа и небесной сферы, а также движение самого объекта (самолета, автомобиля, бегуна на дорожке и т.д.) за время экспозиции.

Причиной дефокусирования изображения могут быть: неправильная установка фокуса в УРИ, сферическая абберация зеркала телескопа, как в случае с телескопом «Хаббл» (рис. 4 слева).

Причиной зашумления изображения могут быть: помехи в атмосфере – пыль, водяные капли, снежинки (внешний шум) или помехи, вызванные выходом из строя элементов матрицы ПЗС (внутренние, инструментальные шумы).

### **Типы изображений:**

*цветные* (полноцветные, RGB, rgb) – в каждом пикселе присутствует смесь красного (R), зеленого (G) и синего (B) цветов в некоторой пропорции, что позволяет довольно точно имитировать цвета различными устройствами (цветными мониторами, принтерами и т.д.);

*полутонные* (серые, gray, grayscale) – в каждом пикселе присутствует также смесь R, G и B цветов, но в равном соотношении, т.е.  $R=G=B$ ;

*индексированные* (цветные палитровые, ind) – устаревшая разновидность RGB-изображений, использующая цветовую карту (map);

*бинарные* (черно-белые, ч/б, двоичные, logical, bw), используемые иногда для отображения (но не для обработки) текстов (не путать ч/б и серые изображения!).

**Классы чисел в системе MatLab.** В пакете IPT (Image Processing Toolbox) используются следующие *классы чисел*:

uint8 – беззнаковое (unsigned) целое (int) 8-битовое число, принимающее значения  $0, 1, \dots, 255 = 2^8 - 1$ ;

double – вещественное число с плавающей точкой двойной точности (16 цифр) в диапазоне от  $\approx \pm 10^{-308}$  до  $\approx \pm 10^{308}$  (8 байт на число);

logical – число со значениями 0 или 1 (1 байт на число).

**Соглашение о координатах.** Будем использовать  $i$  для обозначения номера столбца, а  $j$  – для номера строки изображения. Ось  $x$  (и номер  $i$ ) направим горизонтально слева направо, а ось  $y$  (и номер  $j$ ) – вертикально вниз, при этом  $i = 1, 2, \dots, N$ ,  $j = 1, 2, \dots, M$ . Будем обозначать иногда (но не всегда) интенсивность на смазанном изображении в непрерывном случае через  $g_y(x)$ , а в дискретном случае – через  $g_j(i)$ , где  $x$  и  $i$  направлены вдоль смаза, а  $y$  и  $j$  – перпендикулярно смазу.

Аналогично, будем обозначать интенсивность на дефокусированном изображении (а иногда и на смазанном изображении) в непрерывном случае через  $g(x, y)$ , а в дискретном случае – через  $g(j, i)$ . При этом под дискретным случаем будем подразумевать представление изображения в виде набора пикселей.

Разночтение в обозначениях:  $g(x, y)$  и  $g(j, i)$ , где  $x$  и  $i$  (а также  $y$  и  $j$ ) имеют одинаковое направление, обусловлено тем, что в непрерывном случае принято первым аргументом ставить горизонтальную ось, а в дискретном случае – номер строки  $j$  [1, с. 101], [3, с. 70], [5, с. 102], [7, с. 42].

### 3. Обработка изображений

Рассмотрим прямые и обратные задачи моделирования (формирования) и восстановления искаженных (смазанных, дефокусированных, зашумленных) изображений. Будем использовать математический и программный аппарат.

#### 3.1. Моделирование смазывания изображений (прямая задача)

Рассмотрим данную задачу на примере фотоснимка, смазанного (сдвинутого, смещенного) за время экспозиции. Полагаем, что другие типы искажений (царапины, аберрации и т.д.) отсутствуют или устранены.

**Прямая задача.** Будем считать фотографируемый объект плоским вследствие его удаленности. Пусть объект и фотопленка (или матрица) фотоаппарата расположены параллельно апертуре тонкой линзы фотоаппарата по разные стороны от линзы на расстояниях от нее  $f_1$  и  $f_2$  соответственно (рис. 5).

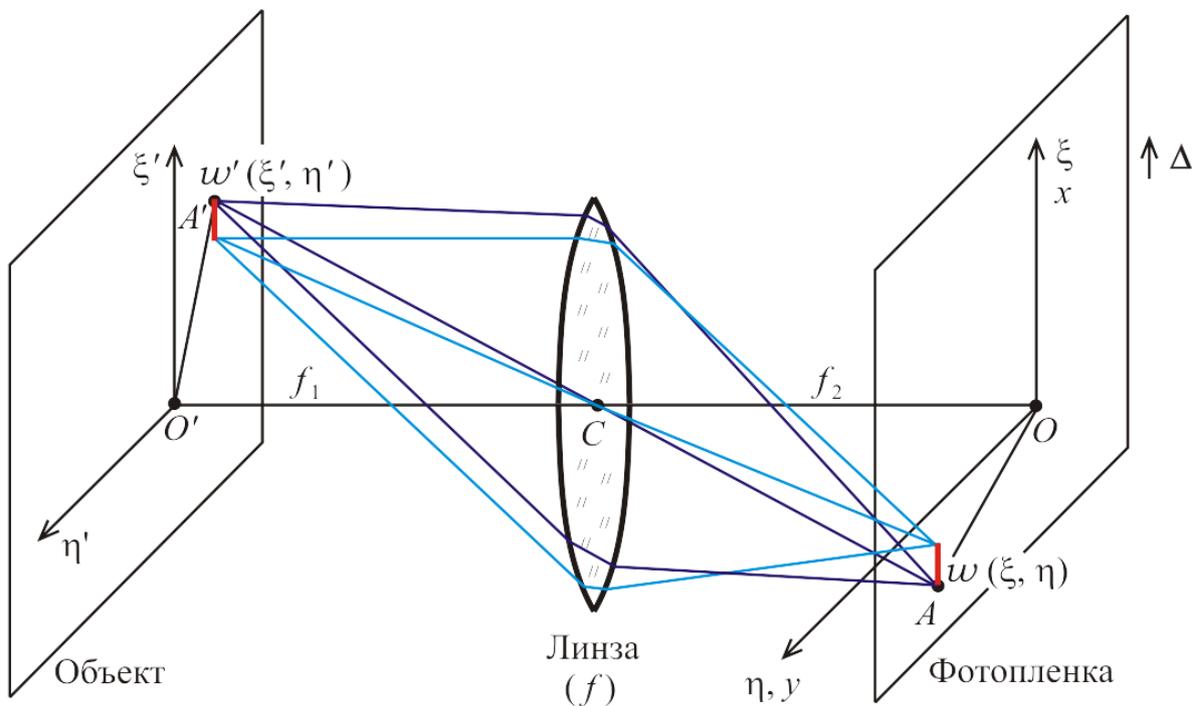


Рис. 5. Схема получения смазанного изображения

При этом должно выполняться известное условие [1, с. 74]:

$$\frac{1}{f_1} + \frac{1}{f_2} = \frac{1}{f}, \quad (9)$$

из которого следует, что фотопленка должна устанавливаться на расстоянии

$$f_2 = \left( \frac{1}{f} - \frac{1}{f_1} \right)^{-1}, \quad (10)$$

в этом случае снимок будет «в фокусе» (четким).

**Пример.** Пусть  $f_1 = 7$  м,  $f = 5$  см, тогда  $f_2 = 5.03$  см согласно (10), а отношение  $q = f_1/f_2 = 139.2$ , т.е. изображение будет в  $q = 139.2$  раза меньше объекта (и перевернуто).

Далее полагаем, что за время экспозиции фотоаппарат с пленкой совершил прямолинейный и равномерный *сдвиг* (смещение) на величину  $\Delta$  под некоторым углом  $\theta$  или сдвиг совершил сам объект (например, движущаяся цель) на величину  $q\Delta$ . Направим ось  $\xi$  вдоль смаза (сдвига), а  $\eta$  – перпендикулярно ему. В результате изображение на фотопленке будет смазанным (сдвинутым, смещенным) вдоль  $\xi$  (см. рис. 6). Для моделирования смаза изображения разработана m-функция `smearing.m` [1, 3].

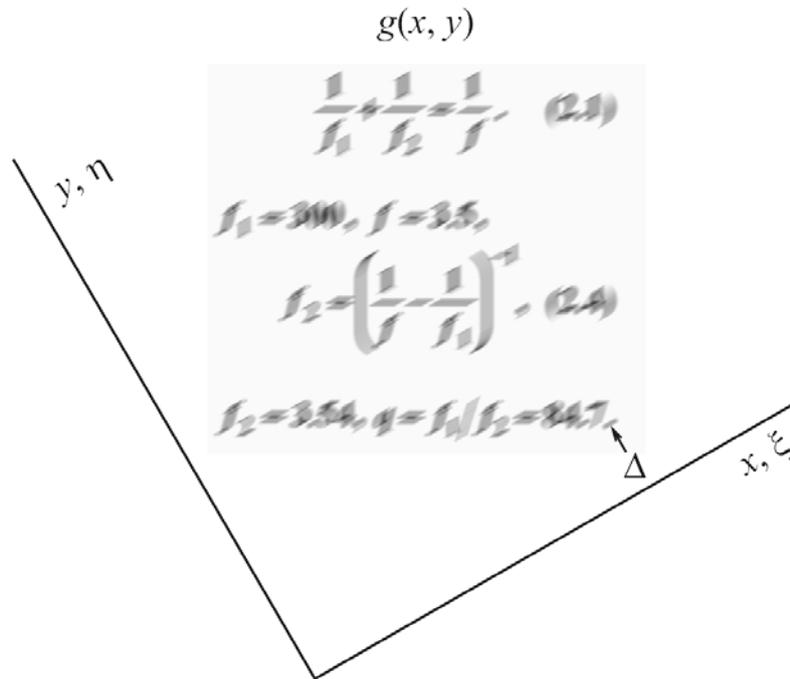


Рис. 6. Пример смазанного изображения.

$\Delta$  – величина смаза (сдвига, смещения), ось  $\xi$  направлена вдоль смаза

**З а м е ч а н и е .** В пакете IPT есть m-функции `fspecial.m` и `imfilter.m` [6, 8], в которых ось  $x$  расположена горизонтально, а  $y$  – вертикально вниз, т.е. оси не связаны с направлением смаза. Мы далее будем пользоваться обоими вариантами – ось  $x$  направлена вдоль смаза и ось  $x$  направлена горизонтально.

Опишем математически прямую задачу. Направим ось  $x$  (и  $\xi$ ) вдоль смаза, а  $y$  (и  $\eta$ ) – перпендикулярно смазу, как на рис 5 и 6. На некоторую точку  $(x, y)$  фотопленки за время экспозиции спроектируется непрерывный набор точек с абсциссами от  $\xi = x$  до  $\xi = x + \Delta$  и с различными интенсивностями  $w(\xi, y)$ . Результирующая интенсивность в некоторой точке  $(x, y)$  фотопленки будет равна сумме (интегралу) интенсивностей  $w(\xi, y)$ ,  $\xi \in [x, x + \Delta]$  в непрерывном случае, когда  $\Delta$  полагается, вообще говоря, нецелым (не в пикселах):

$$g_y(x) = \frac{1}{\Delta} \int_x^{x+\Delta} w_y(\xi) d\xi, \quad (11)$$

где запись сделана построчно – вдоль каждой  $y$ -строки, причем  $y$  играет роль параметра.

В (11) перед интегралом поставлен множитель  $1/\Delta$ . Это связано с тем, что при  $\Delta \rightarrow 0$  (в отсутствие смаза) имеем согласно (11):  $g_y(x) \rightarrow 0/0$ , т.е. неопределенность типа  $0/0$ . Раскроем неопределенность по правилу Лопиталя. Получим:  $g_y(x) \rightarrow w_y(x)$ , т.е. проставление множителя  $1/\Delta$  оправдано.

В дискретном виде интеграл в (11) заменяем конечной суммой с шагом 1 пкс, а  $\Delta$  полагаем целым, например, 20 пкс. Получим:

$$g_j(i) = \frac{1}{\Delta+1} \sum_{k=i}^{i+\Delta} w_j(k), \quad i=1, \dots, n, \quad j=1, \dots, m, \quad (12)$$

где  $j$  – номер строки,  $i$  (и  $k$ ) – номер столбца,  $m$  и  $n$  – число строк и столбцов на дискретизированном изображении соответственно.

На рис. 7 приведены неискаженное и смазанное изображения самолета.



а



б

Рис. 7. а – неискаженное изображение самолета  $w$   $478 \times 744$  пкс; б – смазанное изображение самолета на взлете  $g$ ,  $\Delta = 20$  пкс,  $\theta = 30^\circ$

Рис. 7б показывает, что в результате смазывания не стали видны опознавательные знаки на корпусе самолета. Смазанное изображение получено с помощью  $m$ -функции `smearing.m`, а также `fspecial.m` и `imfilter.m`.

### 3.2. Восстановление смазанных изображений (обратная задача)

Рассмотрим обратную задачу восстановления изображения  $w$  по измеренному смазанному изображению  $g$ . Запишем соотношение (11) иначе:

$$\frac{1}{\Delta} \int_x^{x+\Delta} w_y(\xi) d\xi = g_y(x), \quad (13)$$

а именно, поменяем левую и правую части местами. Однако этого достаточно, чтобы прямая задача, описываемая соотношением (11), трансформировалась в обратную, гораздо более сложную и важную обратную задачу.

Соотношение (13) является основным в задаче устранения смаза изображений. В нем  $g_y(x)$  – измеренное построчно распределение интенсивности на смазанном изображении (на фотопленке) в функции прямоугольных координат  $x, y$ , причем ось  $x$  направлена вдоль смаза,  $\Delta$  – величина смаза, а  $w_y(\xi)$  – искомое распределение интенсивности на неискаженном изображении (той интенсивности, которая была бы на фотопленке в отсутствие смаза, т.е. при  $\Delta = 0$ ).

Соотношение (13) есть одномерное интегральное уравнение (ИУ) типа Вольтерра I рода относительно  $w_y(\xi)$  при каждом фиксированном значении  $y$ , играющем роль параметра, другими словами, (13) есть *набор (семейство, множество) одномерных интегральных уравнений*.

Однако уравнение (13) является неклассическим и нестандартным интегральным уравнением, так как оно не имеет ядра, оба предела интегрирования переменны и т.д., поэтому оно неудобно для численного решения. Преобразуем уравнение (13) в другое, классическое и стандартное уравнение, методы решения которого хорошо разработаны.

Запишем ИУ (13) в виде ИУ Фредгольма I рода типа свертки [1–3, 9, 10]:

$$\int_{-\infty}^{\infty} h(x - \xi) w_y(\xi) d\xi = g_y(x), \quad -\infty < x < \infty, \quad (14)$$

где

$$h(x - \xi) = \begin{cases} 1/\Delta, & -\Delta \leq x - \xi \leq 0, \\ 0, & \text{иначе,} \end{cases} \quad (15)$$

или

$$h(x) = \begin{cases} 1/\Delta, & -\Delta \leq x \leq 0, \\ 0, & \text{иначе.} \end{cases} \quad (16)$$

Здесь  $h$  – математически ядро ИУ, а физико-технически – *функция рассеяния точки* (ФРТ), или *point spread function* (PSF) [4, 5, 8] – это то, во что каждая

точка объекта превращается на изображении при смазывании. В задаче смазывания функция  $h$ , как правило, является разностной, или пространственно-инвариантной, что означает, что величина смаза  $\Delta$  одинакова во всех точках изображения.

**З а м е ч а н и е.** Для устранения смаза может быть использовано также двумерное интегральное уравнение Фредгольма I рода типа свертки [9, 10]

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x - \xi, y - \eta) w(\xi, \eta) d\xi d\eta = g(x, y), \quad -\infty < x, y < \infty, \quad (17)$$

причем ось  $x$  (и  $\xi$ ) направлена горизонтально слева направо, а ось  $y$  (и  $\eta$ ) – вертикально вниз и оси не связаны с направлением смаза изображения. В этом случае при дискретизации задачи функция рассеяния точки  $h$  будет PSF-квазидиагональной матрицей в виде узкой полоски [1, с. 112], [9] (рис. 8).

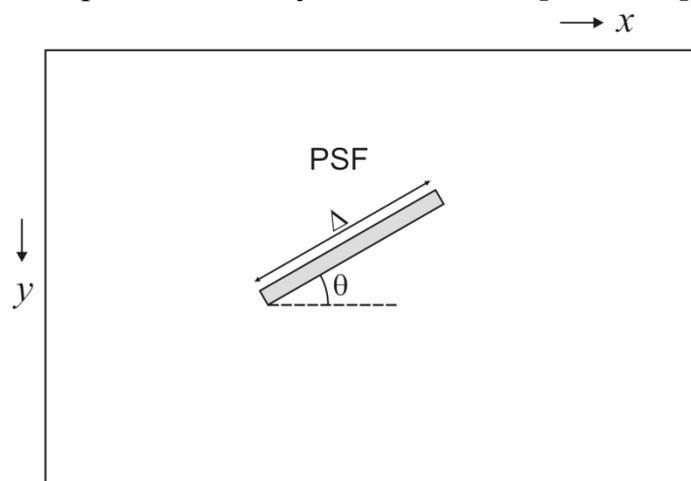


Рис. 8. Функция рассеяния точки (PSF) в виде узкой полоски

Это нашло отражение в IPT-пакете m-функций системы MatLab, где вычисление  $h(x, y)$  выполняется с помощью m-функции `fspecial.m`. В нижеследующем примере преставлен случай смаза длиной  $\Delta = 7$  пкс под углом  $\theta = 30^\circ$ , а также для сравнения под нулевым углом [1, с. 112], [8, с. 181]:

```
>> format short
>> PSF30 = fspecial('motion',7,30)
PSF30 =
    0         0         0         0         0    0.0370    0.0495
    0         0         0    0.0185    0.0876    0.1197    0.0495
    0    0.0000    0.0691    0.1382    0.0691    0.0000         0
    0.0495    0.1197    0.0876    0.0185         0         0         0
    0.0495    0.0370         0         0         0         0         0
>> PSF0 = fspecial('motion',7,0)
PSF0 =
    0.1429    0.1429    0.1429    0.1429    0.1429    0.1429    0.1429
```

Используя такую PSF-матрицу, можно решать обратную задачу устранения смаза изображения с помощью двумерного ИУ (17). Мы будем для устранения смаза изображения использовать как ИУ (14), так и (17).

Рассмотрим некоторые методы решения ИУ (14).

**Метод инверсной фильтрации.** В соответствии с теоремой о свертке [2],

$$H(\omega)W_y(\omega) = G_y(\omega), \quad (18)$$

где  $H(\omega)$ ,  $W_y(\omega)$  и  $G_y(\omega)$  – преобразования Фурье (спектры) функций, входящих в (14):

$$H(\omega) = \int_{-\infty}^{\infty} h(x) e^{i\omega x} dx, \quad -\infty < \omega < \infty, \quad (19)$$

$$W_y(\omega) = \int_{-\infty}^{\infty} w(\xi) e^{i\omega \xi} d\xi, \quad -\infty < \omega < \infty, \quad (20)$$

$$G_y(\omega) = \int_{-\infty}^{\infty} g_y(x) e^{i\omega x} dx, \quad -\infty < \omega < \infty. \quad (21)$$

Решение ИУ (14) в виде обратного преобразования Фурье (ОПФ):

$$w_y(\xi) = \frac{1}{2\pi} \int_{-\infty}^{\infty} W_y(\omega) e^{-i\omega \xi} d\omega, \quad (22)$$

где, в соответствии с (18),

$$W_y(\omega) = \frac{G_y(\omega)}{H(\omega)}. \quad (23)$$

Однако [1, с 121] спектр  $G_y(\omega)$  обычно зашумленного изображения  $g(x)$  при  $\omega \rightarrow \infty$  стремится к некоторой константе («белому шуму»), а спектр  $H(\omega) \rightarrow 0$  при  $\omega \rightarrow \infty$  и имеет нули при конечных  $\omega$ . Это говорит о том, что решение методом инверсной фильтрации согласно (19)–(23) неустойчиво. Более точно, задача решения уравнения (14) является *некорректной* [1–3, 11].

**Метод псевдоинверсной фильтрации.** Для снижения неустойчивости метода инверсной фильтрации можно использовать так называемый *метод псевдоинверсной фильтрации*. Его суть связана с тем, что обычно функция  $g_y(x)$  измеряется для дискретных значений  $x$  с шагом  $\Delta x = \text{const}$ . В этом случае максимальная Фурье-частота равна  $\omega_{\max} = \pi/\Delta x$  – частота Котельникова–Найквиста. Поэтому решение (22) можно записать как

$$w_y(\xi) = \frac{1}{2\pi} \int_{-\omega_{\max}}^{\omega_{\max}} W_y(\omega) e^{-i\omega \xi} d\omega. \quad (24)$$

Такое решение обладает некоторой устойчивостью, так как  $\omega_{\max}$  конечна. Однако остается недостаток – нули функции  $H(\omega)$ . Поэтому даже в отсутствие шума нельзя надеяться на эффективность метода псевдоинверсной фильтрации.

**Метод преобразования Фурье и регуляризации Тихонова.** Гораздо более эффективным является решение ИУ (14) *методом ПФ с регуляризацией Тихонова*. Дадим его краткое изложение применительно к задаче восстановления изображения. Решение уравнения (14) методом ПФ с регуляризацией Тихонова

имеет вид [1–3, 5, 6, 9, 10, 13, 14] (*одномерный вариант*):

$$w_{\alpha y}(\xi) = \frac{1}{2\pi} \int_{-\infty}^{\infty} W_{\alpha y}(\omega) e^{-i\omega\xi} d\omega, \quad (25)$$

где (ср. (23))

$$W_{\alpha y}(\omega) = \frac{H(-\omega)G_y(\omega)}{|H(\omega)|^2 + \alpha\omega^{2p}} \quad (26)$$

– регуляризованный спектр решения,  $\alpha > 0$  – параметр регуляризации,  $p \geq 0$  – порядок регуляризации, обычно  $p = 1$  или  $2$ , а для выбора параметра  $\alpha$  разработан ряд способов: принцип невязки, обобщенный принцип невязки и др. [11–15]. Однако в данном пособии мы будем в основном использовать для выбора  $\alpha$  *способ подбора* (способ визуальной оценки решения  $w_\alpha$ ).

**Метод параметрической фильтрации Винера.** Наряду с методом регуляризации Тихонова будем использовать *метод параметрической фильтрации Винера*, согласно которому [1–3, 5, 8] (ср. (25), (26))

$$w_{Ky}(\xi) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{H(-\omega)G_y(\omega)}{|H(\omega)|^2 + K} e^{-i\omega\xi} d\omega, \quad (27)$$

где  $K \geq 0$  – некоторая константа (параметр), равная отношению шум/сигнал по мощности (NSPR) в спектральной области. Такое решение называется *параметрической винеровской фильтрацией*, или *методом параметрической фильтрации Винера*.

Приведенные соотношения (19)–(27), настроенные на решение ИУ (14), записаны для варианта, когда ось  $x$  (и  $\xi$ ) направлена вдоль смаза, а  $y$  – перпендикулярно ему. Для этого варианта нами разработаны m-функции `smearing.m` и `desmearingf.m`. Однако в пакете IPT для обработки смазанных изображений (как и дефокусированных изображений) использован двумерный вариант. Согласно нему, решается уравнение (17) – двумерное ИУ Фредгольма I рода типа свертки. Его решение методом *двумерного ПФ с регуляризацией Тихонова* имеет вид (ср. (25), (26)):

$$w_\alpha(\xi, \eta) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_\alpha(\omega_1, \omega_2) e^{-i(\omega_1\xi + \omega_2\eta)} d\omega_1 d\omega_2, \quad (28)$$

$$W_\alpha(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)G(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2 + \alpha(\omega_1^2 + \omega_2^2)^p}, \quad (29)$$

$$H(\omega_1, \omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y) e^{i(\omega_1 x + \omega_2 y)} dx dy, \quad (30)$$

$$G(\omega_1, \omega_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{i(\omega_1 x + \omega_2 y)} dx dy. \quad (31)$$

В пакете IPT метод двумерного ПФ с регуляризацией Тихонова согласно (17), (28)–(31) реализован в m-функции `deconvreg.m`. Эта функция может быть использована для восстановления как смазанных, так и дефокусированных изображений в рамках *двумерного варианта*.

Аналогично, решение ИУ (17) методом *параметрической фильтрации Винера* (ПФВ) имеет вид (ср. (27)–(29)):

$$w_K(\xi, \eta) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_K(\omega_1, \omega_2) e^{-i(\omega_1\xi + \omega_2\eta)} d\omega_1 d\omega_2, \quad (32)$$

$$W_K(\omega_1, \omega_2) = \frac{H^*(\omega_1, \omega_2)G(\omega_1, \omega_2)}{|H(\omega_1, \omega_2)|^2 + K}. \quad (33)$$

В пакете IPT метод ПФВ реализован в m-функции `deconvwvr.m`.

**Пример.** Ниже приведена MatLab-программа, иллюстрирующая методы псевдоинверсной фильтрации (ПИФ), регуляризации Тихонова (РТ) и параметрической фильтрации Винера (ПФВ) *устранения смаза двумерным вариантом*.

```

%% Example1.m, Сизиков, гол. программа, MatLab.
%Пример. Демонстрация метода псевдоинверсной фильтрации,
%а также методов регуляризации Тихонова и параметрической
%фильтрации Винера в задаче устранения смаза изображения.
clc,clear
%% Чтение файла с изображением:
RGB = imread('girl.jpg'); %256x256x3 uint8
w = rgb2gray(RGB); %256x256 uint8 gray
%% Прямая задача - смазывание изображения:
Delta = 20; %длина смаза в пкс
D1 = 1/(Delta+1); %для ФРТ h
theta = 0; %угол смаза в град
PSF = fspecial('motion',Delta,theta); %PSF (ФРТ) 1x21 double
gu = imfilter(w,PSF,'circular'); %смазан. изображение 256x256 uint8
figure, subplot(2,2,1), imshow(gu), title('Smearred image gu')
%% Обратная задача - устранение смаза изображения
% методом псевдоинверсной фильтрации:
g = double(gu); %перевод в double 256x256
[m,n] = size(g); %m=256, n=256 (число строк и столбцов)
dx = 1; %шаг дискретизации по x (dx = 1 пкс)
ompmx = 2*pi/dx; %частота Котельникова-Найквиста
dom = ompmx/n; %шаг дискретизации по omega
%в соответствии с теоремой Котельникова и частотой Найквиста
%Расчет 1-мерных массивов double h и H, не зависящих от j:
h = zeros(1,n); %выделение памяти под h (ФРТ) 1x256 double
for i = 1:n %цикл по номеру столбца
    if (i>=n-Delta) h(i) = D1; end %ФРТ h(x) с учетом n-периодичности
end %i
H = fft(h); %1x256 double complex H - ДПФ/ВПФ от h (ОПФ)
%Расчет по строкам:
wr = zeros(m,n); %память под wr (реконструирован. изображение)
for j = 1:m %цикл по строкам
    gj = g(j,:); %вектор 1x256 double gj - j-я строка матрицы g
    Gj = fft(gj); %Gj 1x256 double complex - 1-мерное ДПФ/ВПФ от gj
    %Расчет ДПФ/ВПФ решения W(omega) в j-й строке:
    Wj = Gj./H; %Wj 1x256 double complex

```

```

%Расчет решения w(x) в j-й строке:
wj = ifft(Wj); %1-мерное ОБПФ wj 1x256 double complex
%Внесение вектора abs(wj) в j-ю строку матрицы wd:
wr(j,:) = abs(wj); %неустойчивое решение 256x256 double
end %j
subplot(2,2,2), imshow(uint8(wr)), title('Restored image wr')
%или
wreg0 = deconvreg(gu,PSF,[],0); %неустойчивое решение 256x256 uint8
subplot(2,2,3), imshow(wreg0), title('Restored image wreg0')
%Для сравнения решение методом регуляризации Тихонова:
Tikh = deconvreg(gu,PSF,[],1e-4); %устойчивое решение 256x256 uint8
subplot(2,2,4), imshow(Tikh), title('Restored image Tikh and Wnr')
%A также методом параметрич. фильтрации Винера:
%Wnr = deconvwnr(gu,PSF,1e-4); %устойчивое решение 256x256 uint8
%subplot(2,2,4), imshow(Wnr), title('Restored image Wnr')

```

На рис. 9 приведены результаты работы программы Example1.m.

Из рис. 9 видно, что метод псевдоинверсной фильтрации (ПИФ) дает очень неустойчивое восстановление изображения (см. 2) и 3)), а методы регуляризации Тихонова (РТ) и параметрической фильтрации Винера (ПФВ) дают вполне удовлетворительное восстановление (см. 4)). Заметим, что здесь не учтены шумы. Влияние шумов см. в п. 3.5.

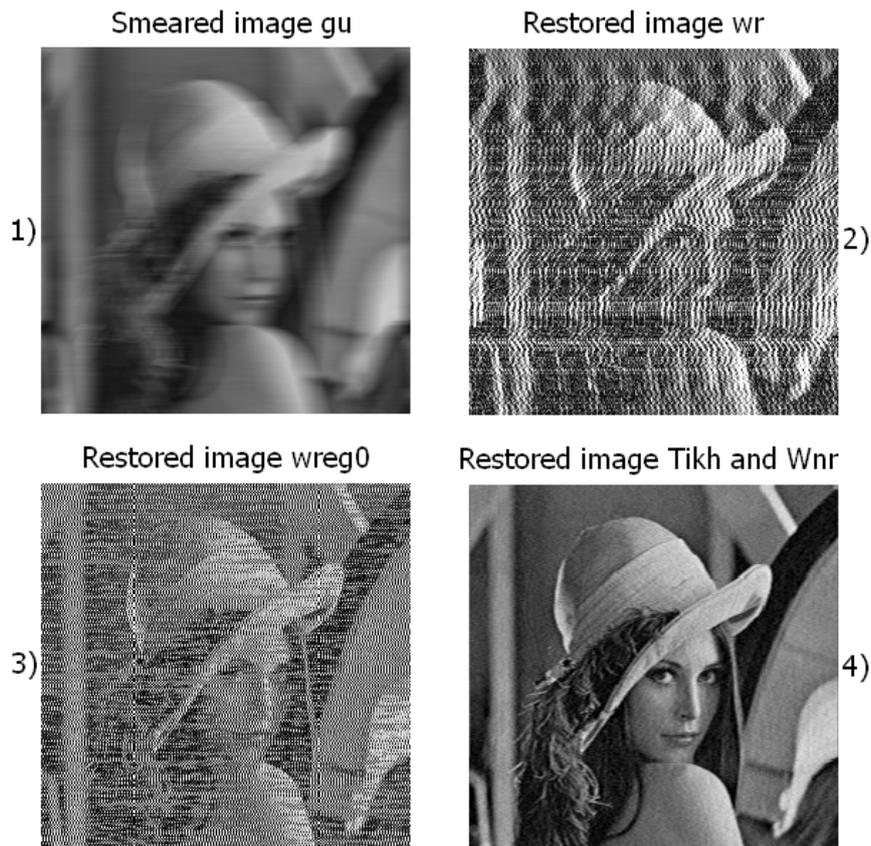


Рис. 9. 1) смазанное изображение girl.jpg 256×256,  $\Delta = 20$ ,  $\theta = 0$ ;  
 2) изображение, восстановленное методом ПИФ согласно (14), (16), (19)–(24);  
 3) изображение, восстановленное методом РТ по deconvreg при  $\alpha = 0$ ;  
 4) изображение, восстановленное методом РТ по deconvreg и  
 методом ПФВ по deconvwnr при  $\alpha = K = 10^{-4}$

**Обработка цветных изображений.** Рассмотрим важный случай, когда изображение является *цветным*.

В пакете IPT системы MatLab цветные изображения представлены в виде RGB (Red, Green, Blue) изображений. Обработка цветных изображений обычно осуществляется следующими двумя способами.

В *первом способе (покомпонентной обработки)* изображение разбивается на три компоненты R, G и B, решается прямая задача – внесение искажений в каждую компоненту, затем решается обратная задача – устранение искажений каждой компоненты некоторым устойчивым методом, например, методом регуляризации Тихонова или методом параметрической фильтрации Винера и в заключение три компоненты соединяются в единое RGB-изображение, которое можно выполнить с помощью m-функции `cat.m`, имеющейся в пакете IPT.

Во *втором способе (векторной обработки)* также решаются прямая и обратная задачи, но без разбиения изображения на три компоненты, а именно, изображение обрабатывается как единое целое (цветные пиксели рассматриваются как трехмерные массивы).

На рис. 10 – обработка цветного изображения первым способом, а на рис. 11 – обработка цветного изображения вторым способом.

Рисунки 10 и 11 показывают хорошее восстановление цветного смазанного изображения, но первый способ требует больше операций, чем второй способ.

### 3.3. Моделирование дефокусирования изображений (прямая задача)

Рассмотрим еще одну важную задачу обработки изображений – задачу моделирования дефокусирования и устранения (реконструкции) дефокусированных изображений (фотоснимков текста, человека, природы, телескопических снимков космических объектов – планет, туманностей, галактик, снимков наземных объектов со спутников, изображений биологических объектов, наблюдаемых в микроскоп – вирусов, ДНК, томографических изображений и т.д.).

Дефокусирование изображения может возникнуть вследствие неправильной установки фокуса фотоаппарата (расфокусировка) или в результате прохождения изображения через атмосферу с турбулентностью [5, с. 384] или из-за наличия сферической аберрации у зеркала телескопа, как это имело место у космического телескопа «Хаббл» (эффekt, аналогичный расфокусировке).

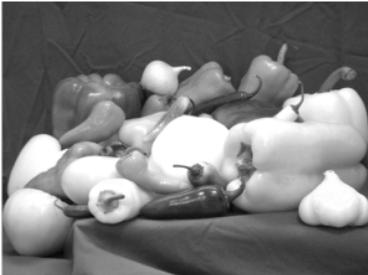
**Прямая задача.** Рассмотрим *прямую задачу* (моделирование дефокусирования) и *обратную задачу* (устранение дефокусирования математическим путем) на примере фотоаппарата. Задача дефокусирования имеет много общего с предыдущей задачей (смазывание изображений), но имеет и отличия.

Пусть снимаемый объект (полагаемый плоским из-за его удаленности) и фотопленка (или матрица сенсоров) расположены параллельно тонкой линзе по разные стороны от нее. Пусть  $f_1$  – расстояние от линзы до объекта, а  $f_2$  – то расстояние от линзы до фотопленки (идеальной фотопленки, установленной в «фокусе»), которое удовлетворяет соотношению (10) и при котором на фотопленке будет четкое изображение объекта.

w - initial RGB-image



wR - R-component



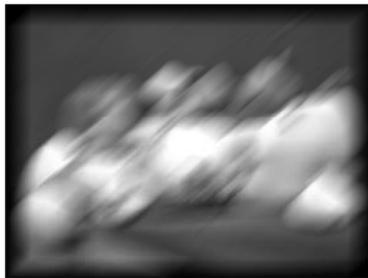
wG - G-component



wB - B-component



gRir - smeared R-component



gGir - smeared G-component



gBir - smeared B-component



wshiftR,  $\alpha=1e-7$



wshiftG,  $\alpha=1e-7$



wshiftB,  $\alpha=1e-7$



wshiftRGB,  $\alpha=1e-7$



Рис. 10. Обработка изображения peppers.png (перцы) первым способом,

метод регуляризации Тихонова, m-функция desmearing.m [1, 3]

smeard RGB-image 384x512x3    restored RGB-image 384x512x3,  $K=1e-3$



Рис. 11. Слева – смазанное цветное изображение,  $\Delta = 50$  пкс,  $\theta = 41^\circ$ ; справа – восстановленное изображение, метод параметрической фильтрации Винера, m-функция deconvnr.m,  $K = 10^{-3}$

Однако мы полагаем, что расстояние от линзы до фотопленки равно не  $f_2$ , а  $f_2 + \delta$ , где  $\delta$  – погрешность фокусировки изображения, т.е. реальная фотопленка установлена «не в фокусе» (рис. 12).

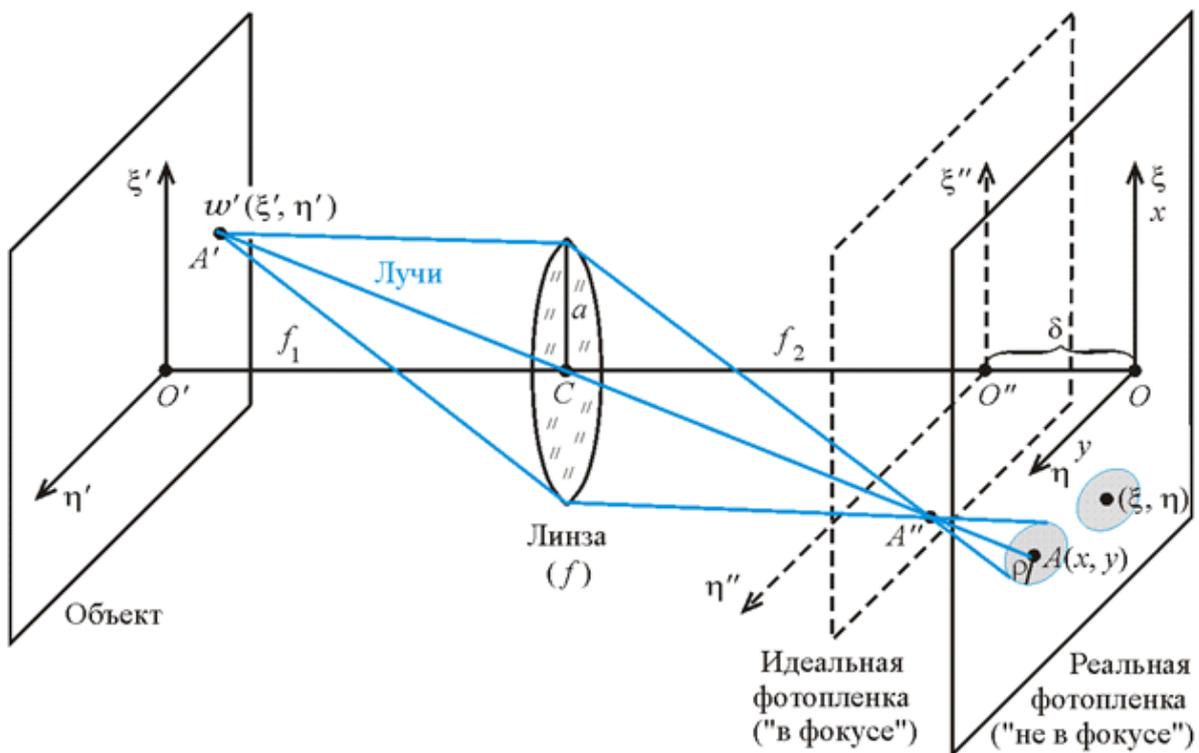


Рис. 12. Схема получения дефокусированного изображения

Как видно из рис. 12, лучи из точки  $A'$  после их прохождения через линзу отобразятся на реальной фотопленке не в точку, а в некоторое размытое пятно, называемое *функцией рассеяния точки* – ФРТ (point spread function – PSF [16]).

Сначала рассмотрим *простейший случай*, когда каждое такое пятно (т.е.

ФРТ) есть *однородный круг* радиуса  $\rho$ , что соответствует пространственно-инвариантной ФРТ (на рис. 12 отображен именно этот простейший случай). Заметим, что в  $m$ -функции `fspecial.m` пакета IPT системы MatLab этому случаю соответствует вариант получения ФРТ: `h=fspecial('disk',radius)` [8, с. 115].

**Пример.** Ниже приведен текст MatLab-программы, дающей пример дефокусирования изображения, полученного из исходного изображения `text.bmp`:

```
%Example2dir.m, головная программа, Сизиков, MatLab.
%Формирование текстового изображения, дефокусированного с помощью ФРТ
%в виде однородного диска радиуса rho (прямая задача, direct problem).
clc,clear
%Чтение файла с текстовым изображением:
RGB = imread('text.bmp'); %618x690x3 uint8 (текст)
w = double(rgb2gray(RGB)); %618x690 double
%PSF (ФРТ, или маска) дефокусирования изображения:
rho = 10; %радиус однородного диска = 10 пкс
PSF = fspecial('disk',rho);
%Дефокусирование изображения (image defocusing, image blurring):
defoc = imfilter(w,PSF,'symmetric'); %дефокусированное изображение
figure
subplot(1,2,1), imshow(uint8(w)), title('w 618x690','FontSize',16)
subplot(1,2,2), imshow(uint8(defoc)),
title('defoc 618x690, \rho=10','FontSize',16)
```

На рис. 13 – результат работы данной программы.

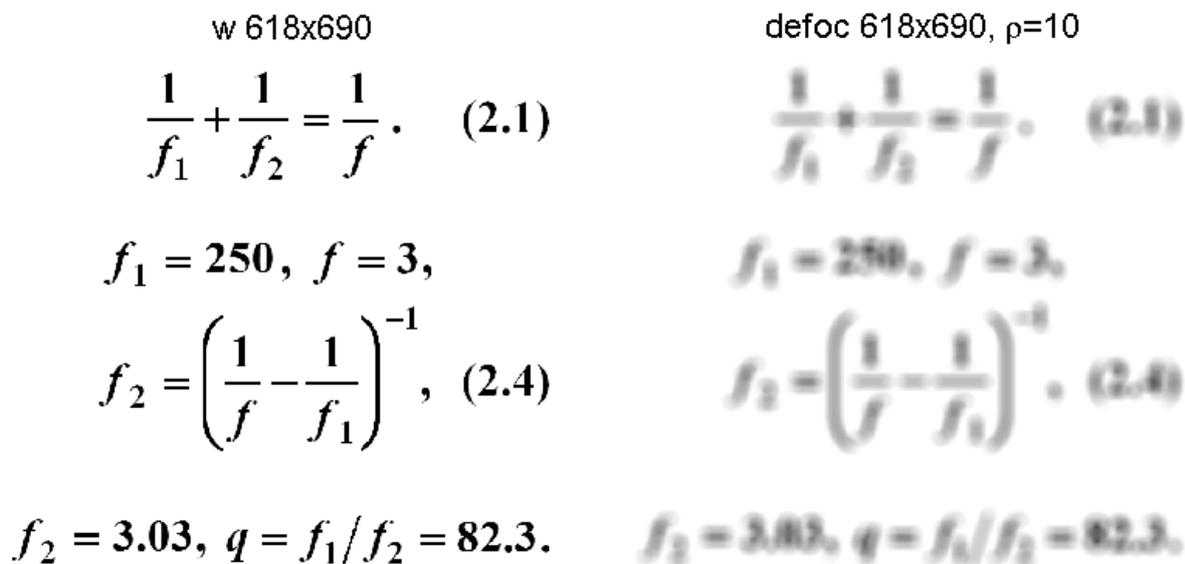


Рис. 13. Пример.  $w$  – исходное неискаженное изображение текста;  $defoc$  – дефокусированное изображение, ФРТ – однородный круг,  $\rho=10$

Видим, что точки на изображении (всего шесть точек) действительно превратились в однородные круги (диски) одинакового радиуса.

### 3.4. Восстановление дефокусированных изображений (обратная задача)

**Описание обратной задачи посредством интегрального уравнения.** Опишем математически задачу дефокусирования в простейшем случае, когда точка  $A'$  отображается на реальной фотопленке в однородный круг радиуса

$$\rho = \frac{a\delta}{f_2} \quad (34)$$

с центром в точке  $A(x, y)$  (см. рис. 12), где  $a$  – радиус апертуры линзы (апертура – действующее отверстие оптической системы).

Рассмотрим, помимо круга с центром в точке  $A(x, y)$ , другой круг с центром в точке  $(\xi, \eta)$  (см. рис. 12). Радиусы этих кругов одинаковы и равны  $\rho$  (см. (34)), а площади кругов равны  $S = \pi\rho^2$ . В результате интенсивность  $w(\xi, \eta)$  будет «размазана» по кругу радиуса  $\rho$  и площади  $S = \pi\rho^2$  с постоянной плотностью интенсивности  $w(\xi, \eta)/\pi\rho^2$ .

Интенсивность в точке  $A(x, y)$  будет результатом суммирования (интегрирования) по всем тем кругам, которые накрывают точку  $A(x, y)$ . Условие накрытия точки  $A(x, y)$  кругом с центром в точке  $(\xi, \eta)$  и радиусом  $\rho$  есть

$$\sqrt{(x - \xi)^2 + (y - \eta)^2} \leq \rho. \quad (35)$$

В результате интенсивность в точке  $A(x, y)$  будет равна

$$g(x, y) = \iint_{\sqrt{(x-\xi)^2+(y-\eta)^2} \leq \rho} \frac{w(\xi, \eta)}{\pi\rho^2} d\xi d\eta. \quad (36)$$

Уравнение (36) есть двумерное интегральное уравнение I рода относительно  $w(\xi, \eta)$ . Однако оно имеет *нестандартную форму*: вместо пределов интегрирования стоит неравенство (35), а также в явном виде нет ядра (точнее, оно равно константе  $1/\pi\rho^2$ ). Нестандартность уравнения (36) создает сложности с его решением. Преобразуем его к стандартной форме. Запишем (36) в виде [1–3, 9, 13–15, 17, 18] (ср. (17)):

$$\int_{-\infty - \infty}^{\infty} \int_{-\infty}^{\infty} h(x - \xi, y - \eta) w(\xi, \eta) d\xi d\eta = g(x, y), \quad (37)$$

где

$$h(x, y) = \begin{cases} \frac{1}{\pi\rho^2}, & \sqrt{x^2 + y^2} \leq \rho, \\ 0, & \text{иначе.} \end{cases} \quad (38)$$

Соотношение (37) есть двумерное интегральное уравнение Фредгольма I рода типа свертки. В нем  $g(x, y)$  – интенсивность на дефокусированном фотоснимке,  $w(\xi, \eta)$  – искомая интенсивность на неискаженном снимке (на снимке, который был бы получен при  $\delta = 0$ ), а  $h(x, y)$  есть ядро интегрального уравнения, являющееся функцией рассеяния точки (ФРТ) в простейшем случае, когда ФРТ – однородный круг постоянного радиуса  $\rho$ .

При этом значение  $\rho$  можно оценить согласно (34), где  $a$  и  $f_2$  известны, а  $\delta$  можно оценить путем подбора.

Если в изображении имеются точки, как на левом рис. 13, то  $\rho$  можно оценить по наименьшему пятну на дефокусированном изображении, как на правом рис. 13.

Но одним из наиболее эффективных и универсальных является развиваемый в последнее время так называемый *спектральный способ* [1, с. 213–220], [19, 20]. Согласно нему, по спектру (двумерному ПФ) дефокусированного изображения можно определить параметр  $\rho$ .

После определения параметра  $\rho$  можно решать ИУ (37). Его решение методом *двумерного ПФ с регуляризацией Тихонова* имеет вид (28)–(31).

В пакете IPT имеется m-функция `deconvreg.m`, а нами разработана m-функция `refocusingT.m` для восстановления дефокусированных изображений путем решения двумерного ИУ (37) (или (17)) методом двумерного ПФ с регуляризацией Тихонова согласно (28)–(31).

ИУ (37) (или (17)) можно решать также методом *параметрической фильтрации Винера* согласно (32)–(33). В пакете IPT имеется m-функция `deconvwnr.m` для восстановления дефокусированных изображений методом параметрической фильтрации Винера (ПФВ).

**Пример.** Ниже приведен текст MatLab-программы `Example2inv`, являющейся продолжением программы `Example2dir` и дающей пример восстановления дефокусированного изображения методом ПФВ:

```
%% Example2inv.m, Сизиков, гол. программа, MatLab.
%Продолжение программы Example2dir.m.
%Восстановление дефокусированного изображения методом
%параметрической фильтрации Винера (ПФВ, deconvwnr)
%(обратная задача, inverse problem).
clc,clear %очистка Command Window и Workspace
%% Чтение файла с текстовым изображением:
RGB = imread('text.bmp'); %618x690x3 uint8 (текст)
w = double(rgb2gray(RGB)); %618x690 double
%PSF (ФРТ, маска) дефокусирования изображения:
rho = 10; %радиус однородного диска = 10 пкс
PSF = fspecial('disk',rho);
%Дефокусирование изображения (прямая задача):
defoc = imfilter(w,PSF,'symmetric'); %дефокусированное изображение
%% Обратная задача - восстановление дефокусированного
%изображения defoc методом ПФВ:
W_K = deconvwnr(defoc,PSF,1e-5); %618x690 double
sig_W_K = rmsd(W_K,w); %относительная погрешность = 0.028
figure
subplot(1,2,1), imshow(uint8(defoc)),
title('defoc 618x690, \rho=10','FontSize',16)
subplot(1,2,2), imshow(uint8(W_K),[]),
title('W_K 618x690, K=1e-5, \sigma=0.028','FontSize',16)
```

На рис. 14 – результат работы программы `Example2inv`. Видим, что восстановление хорошее, а именно относительная погрешность восстановления  $\sigma=0.028=2.8\%$  согласно разработанной m-функции `rmsd.m` [1, 3]. Однако восстановление выполнено в отсутствие шума и с точным  $\rho$ .

<p>defoc 618x690, <math>\rho=10</math></p> $\frac{1}{f_1} + \frac{1}{f_2} = \frac{1}{f}, \quad (2.1)$ <p><math>f_1 = 250, f = 3,</math></p> $f_2 = \left( \frac{1}{f} - \frac{1}{f_1} \right)^{-1}, \quad (2.2)$ <p><math>f_2 = 3.03, q = f_1/f_2 = 82.3,</math></p>	<p>W_K 618x690, <math>K=1e-5, \sigma=0.028</math></p> $\frac{1}{f_1} + \frac{1}{f_2} = \frac{1}{f}, \quad (2.1)$ <p><math>f_1 = 250, f = 3,</math></p> $f_2 = \left( \frac{1}{f} - \frac{1}{f_1} \right)^{-1}, \quad (2.4)$ <p><math>f_2 = 3.03, q = f_1/f_2 = 82.3.</math></p>
--	---

Рис. 14. Пример. defoc – дефокусированное изображение, ФРТ – однородный круг,  $\rho=10$ ; W\_K – изображение, восстановленное методом ПФВ, относительная погрешность восстановления  $\sigma=0.028$

### 3.5. Шумы на изображениях

Изображения обычно бывают зашумлены. Шумы бывают различных типов [1, 3, 5–8, 21].

**Типы шумов.** Рассмотрим шумы следующих типов:

- *внешний* (атмосферный) и *внутренний* (инструментальный);
- *гауссов* и *импульсный*, а также *белый*, *равномерный*, *рэлеевский* и др.,
- *в пространственной области* и *в частотной области*.

*Внешний шум* на изображении возникает, когда между объектом и камерой имеется пыль, туман, капли влаги, снежинки, а *внутренний шум* может быть обусловлен сбоем в работе сенсоров матрицы ПЗС.

*Гауссов шум* обычно вызван непрерывным случайным процессом, например, туманом или дымкой, а также шумом в электронных цепях. Особенность гауссова шума состоит в том, что он присутствует в каждом пикселе изображения и поэтому нередко трудно наблюдать.

*Импульсный шум* обусловлен пылью, каплями дождя (внешними причинами), а также выходом из строя сенсоров матрицы ПЗС, в результате чего возникают так называемые «битые пиксели» (внутренние причины). Импульсный шум обычно имитируется черными и белыми точками на изображении (не в каждом пикселе). Это так называемый *биполярный импульсный шум*. Он хорошо наблюдается на изображении (см. рис. 2б).

*Шум в пространственной области* характеризуется функцией плотности вероятности, а *шум в частотной области* описывается преобразованием Фурье (ПФ).

Мы будем рассматривать в основном шум гауссов и импульсный (внешний и внутренний, в пространственной и частотной областях).

**Способы наложения гауссова шума на изображение.** Чтобы наложить на изображение *гауссов шум*, можно использовать *m*-функцию `imnoise.m`, например,  $g = \text{imnoise}(w, 'gaussian', m, v)$ , где  $w$  – неискаженное изображение,  $m$  – среднее значение шума,  $v$  – его дисперсия, а  $g$  – зашумленное изображение.

Однако обращение к `imnoise.m` с опцией `'gaussian'` связано с некоторыми неудобствами, а именно [8, с. 156], функция `imnoise.m` преобразует изображение в класс `double` в диапазоне  $[0,1]$ , поэтому, чтобы добавить, например, гауссов шум с матожиданием  $m$  и СКО  $s$  к изображению  $f$  класса `uint8`, нужно обратиться к `imnoise` следующим образом:  $g = \text{imnoise}(w, 'gaussian', m/256, (s/256)^2)$ ; Такое обращение весьма неудобно.

Нами разработана более удобная *m*-функция `portmnoise.m`, обращение к которой имеет вид:  $g = \text{portmnoise}(w, \text{per}) + m0$ , где `per` – уровень шума в процентах, а  $m0$  – среднее значение (матожидание) гауссова шума. Далее мы будем неоднократно пользоваться *m*-функцией `portmnoise.m`.

**Способы фильтрации гауссова шума на изображении.** Рассмотрим способы фильтрации (подавления) гауссова шума.

Для *фильтрации гауссова шума*, как показано в [1, 3, 5, 21] и др., весьма эффективными являются среднеарифметический фильтр и адаптивный фильтр Винера. В обоих фильтрах по изображению размером  $M \times N$  скользит *малое окно* размером  $m \times n$  (см. рис. 15).

Обозначим область малого окна через  $S_{xy}$ .

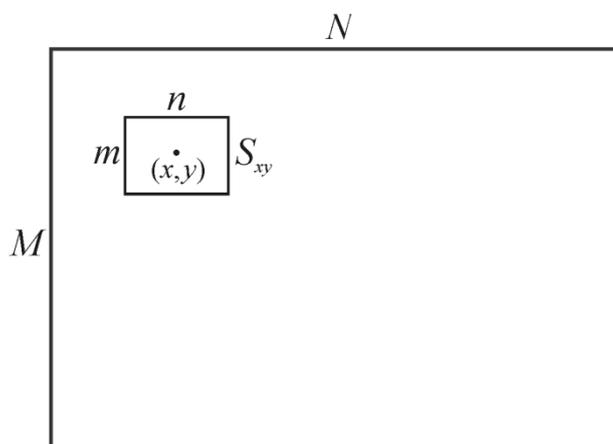


Рис. 15. Изображение  $M \times N$  и скользящее окно  $m \times n$

В *среднеарифметическом фильтре* [5, с. 346] при каждом положении окна вычисляется среднее арифметическое значение интенсивностей точек, попавших в окно  $S_{xy}$ :

$$f(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t), \quad (39)$$

где  $(s, t)$  – точки в окне,  $g(s, t)$  – зашумленные интенсивности,  $(x, y)$  – центральная точка окна, а  $f(x, y)$  – отфильтрованная интенсивность изображения, присваиваемая точке  $(x, y)$ . Среднеарифметический фильтр реализуется следую-

шим образом:  $psf = fspecial('average',[m,n]); f = imfilter(g,psf)$  [8, с. 173].

Идея среднеарифметического фильтра состоит в том, что шумы на изображении имеют частые флуктуации и при усреднении в значительной степени должны себя подавить (сгладить), а изображение изменяется незначительно в пределах умеренного, например,  $5 \times 5$ -окна и себя практически не подавляет.

Усредняющий фильтр 'average' относится к классу фильтров низких частот [6, с. 531], т.е. он подавляет высокие частоты, или мелкомасштабные флуктуации в изображении, сохраняя низкие частоты. Однако его применение приводит к некоторому размытию изображения и понижению его резкости.

Более эффективным является *адаптивный фильтр Винера* (адаптивная фильтрация Винера) [21, р. 536–548], [6, с. 553], [5, с. 357–358]. Этот фильтр называется адаптивным потому, что в нём предусмотрено изменение статистических свойств изображения (матожидания и дисперсии) от точки к точке. Адаптивный фильтр Винера реализован в  $m$ -функции `wiener2.m`.

Обращение к  $m$ -функции `wiener2.m` может быть следующих типов:

1)  $f = wiener2(g)$ ; 2)  $f = wiener2(g, [m n])$ ; 3)  $[f,noise] = wiener2(g,[m n])$ ; 4)  $f = wiener2(g,noise)$ ; 5)  $f = wiener2(g,[m n],noise)$ ;

Здесь  $g$  – зашумленное изображение, матрица;  $[m n]$  – скользящее окно (по умолчанию  $[3 3]$ );  $noise$  – дисперсия шума, скаляр;  $f$  – отфильтрованное изображение, приближение к неискаженному изображению  $w$ , матрица.

**Способы наложения импульсного шума на изображение.** Теперь рассмотрим способы формирования биполярного импульсного шума.

*Биполярный импульсный шум*, называемый также «salt & pepper» («соль и перец»), представляет собой набор черных и белых точек на изображении. Он реализован в  $m$ -функции `imnoise.m` в виде:  $g = imnoise(w,'salt \& pepper',d)$  [6, с. 549], [8, с. 156], достаточно удобной функции, поэтому для зашумления изображения импульсным шумом мы будем использовать  $m$ -функцию `imnoise.m` (здесь  $d \in [0,1]$  – доля зашумленности изображения).

**Способы фильтрации импульсного шума на изображении.** Для фильтрации импульсного шума, как показано в [1, 3, 5, 8], весьма эффективным является *медианный фильтр* (медианная фильтрация изображения) [6, с. 550], [8, с. 119], [21, р. 469], [22, с. 124]. Этот фильтр впервые предложил в 1974 г. Дж. Тьюки – американский математик, один из авторов знаменитого быстрого преобразования Фурье (БПФ), поэтому мы будем называть его *медианным фильтром Тьюки*.

В медианном фильтре Тьюки, как в предыдущих (среднеарифметическом и винеровском) фильтрах, используется скользящее окно (рис. 15). Однако при каждом положении окна вычисляется не среднеарифметическое значение (39) интенсивностей  $g$  точек, попавших в окно (в этом случае будет усреднение как по «беднякам» – малым интенсивностям, так и по «богачам» – большим интенсивностям), а вычисляется так называемое *медианное значение*. Для этого интенсивности  $g$  окна располагаются по возрастанию (всего  $mn$  интенсивностей) и центральной точке окна присваивается интенсивность под средним (медиан-

ным) номером  $(mn + 1)/2$ . В этом случае из игры исключаются «бедняки» и, главное, «богачи», сильно завышающие среднеарифметическое значение.

Медианный фильтр Тьюки реализован в m-функции `medfilt2.m`. Обращения к ней могут быть следующие:

1) `f = medfilt2(g)`; 2) `f = medfilt2(g,[m n])`;

Здесь `g` – зашумленное изображение, матрица; `[m n]` – скользящее окно (по умолчанию `[3 3]`); `f` – отфильтрованное изображение, матрица.

Для иллюстрации изложенных приемов наложения шумов (гауссова и импульсного) на изображения и их фильтрации разработана головная MatLab-программа `Noises.m` (шумы на изображении монет и их фильтрация):

```
%% Noises.m, Сизиков, гол. программа, MatLab.
%Формирование шумов (гауссова и импульсного) на изображениях
%и их фильтрация (подавление).
clc,clear
%% Чтение файла с неискаженным изображением:
w = imread('eight.tif'); %242x308 uint8 gray (монеты)
%% Наложение серых (полутоновых, gray) шумов:
randn('state',0); %инициализация генератора норм. случайных чисел
rand('state',0); %инициализация генератора равном. случайных чисел
g1 = imnoise(w,'gaussian',5/256,(25/256)^2); %добавление гауссова шума
g2 = normnoise(w,10)+5; %добавление 10%-го гауссова шума
g3 = imnoise(w,'salt & pepper',0.05); %добавление импульсного шума
%% Вывод на экран серых зашумленных изображений:
figure
subplot(2,3,1), imshow(g1), title('g1','FontName','Tahoma')
subplot(2,3,2), imshow(g2), title('g2','FontName','Tahoma')
subplot(2,3,3), imshow(g3), title('g3','FontName','Tahoma')
%% Фильтрация шумов:
f1 = wiener2(g1,[5 5]); f2 = wiener2(g2,[5 5]);
f3 = medfilt2(g3,[5 5]);
%% Вывод на экран серых отфильтрованных изображений:
subplot(2,3,4), imshow(f1), title('f1','FontName','Tahoma')
subplot(2,3,5), imshow(f2), title('f2','FontName','Tahoma')
subplot(2,3,6), imshow(f3), title('f3','FontName','Tahoma')
```

На рис. 16 – результат работы программы `Noises.m`.

Рис. 16 показывает, что гауссов шум, присутствующий в каждой точке изображений `g1` и `g2`, наблюдается нечетко, скорее, как однородный фон, а биполярный импульсный шум, присутствующий далеко не в каждой точке изображения `g3` (точнее, в `d`-той части изображения) наблюдается весьма четко.

Рис. 16 показывает также, что гауссов шум хорошо фильтруется адаптивным фильтром Винера, а импульсный шум – медианным фильтром Тьюки. Если же гауссов шум фильтровать медианным фильтром, а импульсный шум – адаптивным фильтром, то, как показано в [1, 3], результаты фильтрации будут заметно хуже.

**Замечание.** Помимо вышеизложенных фильтров шумов на изображениях существует также *ранговый фильтр* (ранговая, или порядковая фильтрация) [6, с. 550], [8, с. 118]. Ранговая фильтрация является развитием медианной фильтрации и позволяет создать очернение (эрозию) и обеливание изображе-

ния. Она реализована в  $m$ -функции `ordfilt2.m`. Отличие ранговой фильтрации от медианной состоит в том, что в ней центральной точке окна можно присваивать интенсивность не обязательно с медианным номером  $(mn + 1)/2$ , но и с меньшим номером (будет очернение) или с бóльшим номером (будет обеливание).

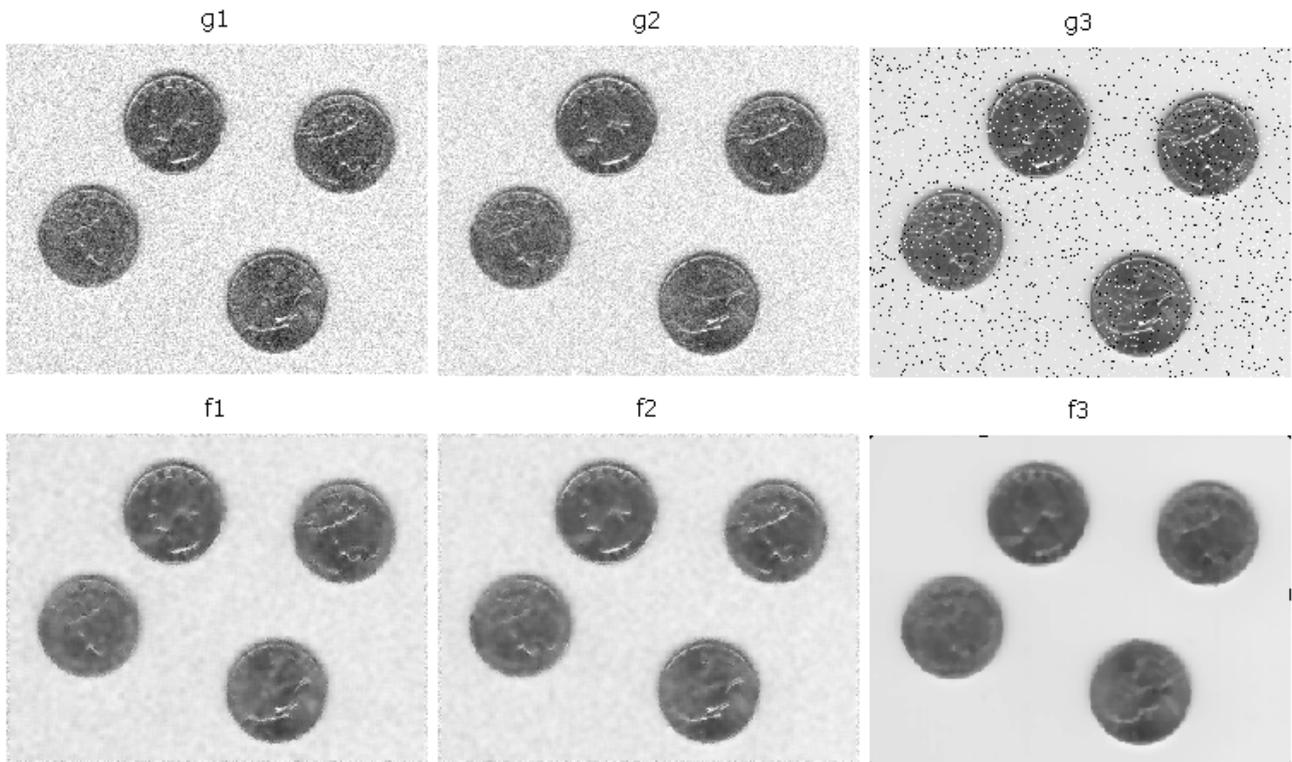


Рис. 16. Верхний ряд – зашумление изображения монет,  $g1$  – гауссов шум,  $m$ -функция `imnoise`,  $g2$  – гауссов шум,  $m$ -функция `poimnoise`,  $g3$  – импульсный шум,  $m$ -функция `imnoise`. Нижний ряд – фильтрация шумов,  $f1$  – гауссов шум,  $m$ -функция `wiener2`,  $f2$  – гауссов шум,  $m$ -функция `wiener2`,  $f3$  – импульсный шум,  $m$ -функция `medfilt2`

**Погрешность восстановления изображения.** Если точное изображение  $\bar{w}$  известно (например, в случае обработки модельного изображения), то можно рассчитать относительную среднеквадратическую погрешность восстановления изображения  $w_\alpha$  методом регуляризации Тихонова [1, 3, 20, 23, 24]:

$$\sigma_{\text{rel}}(\alpha) = \frac{\sqrt{\sum_{j=1}^m \sum_{i=1}^n [(w_\alpha)_{ji} - \bar{w}_{ji}]^2}}{\sqrt{\sum_{j=1}^m \sum_{i=1}^n \bar{w}_{ji}^2}}. \quad (40)$$

Использование формулы (40) позволяет не визуально, а численно оценить погрешность восстановления изображения. Аналогична формула для погрешности восстановления изображения  $w_K$  методом параметрической фильтрации Винера, лишь вместо  $\alpha$  в формуле (40) нужно писать  $K$ .

**Восстановление смазанных и зашумленных изображений.** Рассмотрим случай, когда изображение смазано и зашумлено. В качестве примера рассмотрим смазанное изображение Сатурна с четырьмя спутниками и с наложенным несмазанным внешним импульсным шумом (см. рис. 17а) [24].

Такая ситуация может возникнуть из-за того, что космические объекты часто бывают слабосветящиеся и при их фотографировании требуется продолжительная экспозиция. За это время, если телескоп не вращается вслед за небесной сферой, изображение объекта смажется, а атмосферный шум (пыль, капли влаги) не будет смазан. Если же телескоп вращается, то смазывание изображения Сатурна со спутниками могло произойти из-за рассогласования вращения телескопа и небесной сферы, а импульсный несмазанный (инструментальный) шум возник в результате выхода из строя ряда сенсоров матрицы ПЗС.

Разработана программа `Saturn.m` для обработки изображения, представленного на рис. 17а:

```
%% Saturn.m, Сизиков, гол. программа, MatLab.
%Сатурн с четырьмя спутниками.
%Сначала решается прямая задача - моделирование смазанного
%изображения Сатурна и его зашумление импульсным несмазанным шумом.
%Далее решается обратная задача - метод ПФВ без фильтрации шума,
%а также предшествующая фильтрация шума медианным фильтром
%и затем устранение смаза методом ПФВ.
clc,clear
%% Чтение файла с изображением:
RGB = imread('saturn.png'); %1500x1200x3 uint8
w = rgb2gray(RGB); %1500x1200 uint8
w = double(w); %1500x1200 double
%% Прямая задача - смазывание изображения (image smearing):
Delta = 40; %длина смаза в пкс
theta = 35; %угол смаза в град
PSF = fspecial('motion',Delta,theta); %PSF (ФПТ) 25x33 double
c = imfilter(w,PSF,'circular'); %смаз. изображение 1500x1200 double
%% Добавление импульсного несмазанного шума к смазанному изображению:
rand('state',0); %установка генераторов случайных
randn('state',0); %чисел в начальное состояние
d = 0.02; %доля зашумленных пикселей (2%)
cn = imnoise(uint8(c),'salt & pepper',d); %1500x1200 uint8
figure, subplot(2,2,1), imshow(cn,[]), title('(а)')
%% Обратная задача - устранение смаза (removal of smear)
%методом ПФВ без фильтрации шума:
Wnr_cn = deconvwnr(cn,PSF,1e-1); %1500x1200 uint8
sig_relWnr_cn = rmsd(Wnr_cn,w); %=0.192 (отн. погрешность)
subplot(2,2,2), imshow(Wnr_cn,[]), title('(б)')
%% Предварительная медианная фильтрация импульсного шума:
c3 = medfilt2(cn,[3 3]); %1500x1200 uint8
sig_relc3 = rmsd(c3,w); %=0.115
subplot(2,2,3), imshow(c3,[]), title('(в)')
%% Устранение смаза методом ПФВ после фильтрации шума:
Wnr_c3 = deconvwnr(c3,PSF,1e-3); %1500x1200 double
sig_relWnr_c3 = rmsd(Wnr_c3,w); %=0.0506
subplot(2,2,4), imshow(Wnr_c3,[]), title('(г)')
```

На рис. 17 – результат работы программы `Saturn.m`.

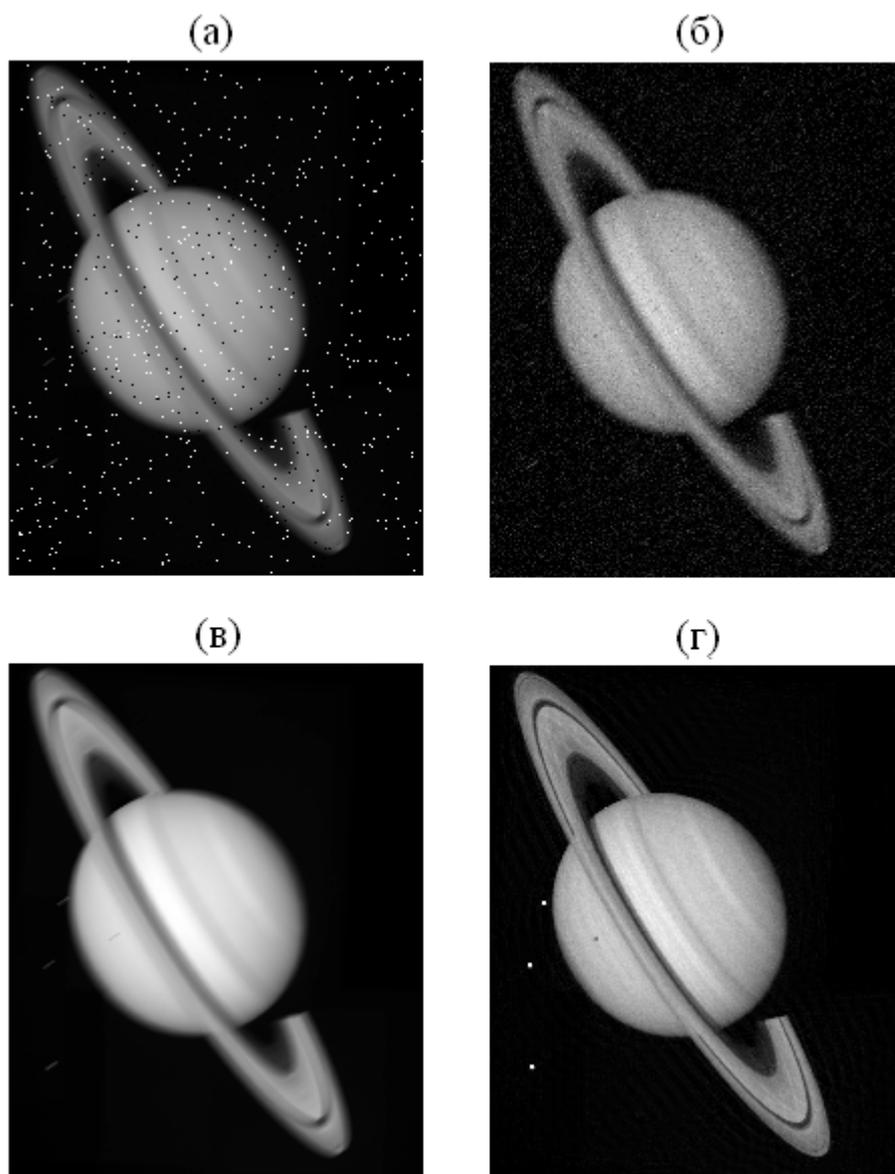


Рис. 17. а – смазанное изображение Сатурна со спутниками и с несмазанным импульсным шумом; б – обработка изображения (а) методом ПФВ без фильтра,  $K = 10^{-1}$ ,  $\sigma_{\text{rel}} = 0.192$ ; в – предшествующий медианный фильтр изображения (а) с маской  $3 \times 3$ ; г – ПФВ изображения (в),  $K = 10^{-3}$ ,  $\sigma_{\text{rel}} = 0.050$

С помощью данной программы смоделировано смазывание изображения Сатурна и спутников на величину смаза  $\Delta = 40$  пкс под углом  $\theta = 35^\circ$  и наложен на изображение импульсный несмазанный шум (рис. 17а). Рис. 17а показывает, что за шумом совершенно не видно смазанных (в виде очень слабых штрихов) изображений спутников, хотя они присутствуют. Далее (рис. 17б) выполнено устранение смаза изображения (а) методом параметрической фильтрации Винера (ПФВ) без предварительной фильтрации шума медианным фильтром в надежде, что метод ПФВ сам устранил не только смаз, но и шум. Видим, что результат получился неудовлетворительный ( $\sigma_{\text{rel}} = 0.192$ ). Для улучшения результата выполнена предшествующая медианная фильтрация шу-

ма на изображении (а). Рис. 17в показывает, что шум отфильтровался и остался лишь смаз как изображения Сатурна, так и спутников (в виде слабых штрихов). Затем выполнено устранение смаза методом ПФВ (рис. 17г). Видим, что хорошо восстановились изображения ( $\sigma_{\text{rel}} = 0.050$ ) крупного объекта – Сатурна и, главное, изображения мелких объектов – всех четырех спутников, причем без ложных мелких объектов.

Данный пример говорит о том, что в случае, когда изображение отягощено как смазом, так и шумом, необходимо шум устранять специальными методами (медианным фильтром и т.д.), а смаз устранять методами параметрической фильтрации Винера (ПФВ) или регуляризации Тихонова (РТ) и т.д. При этом, как показано в [24], в случае импульсного шума важно, в какой последовательности нужно выполнять фильтрацию шума медианным фильтром и устранение смаза (в зависимости от конкретной помехо-сигнальной ситуации), а для гауссова шума порядок выполнения этих операций несущественен. Это объясняется тем, что медианная фильтрация (МФ) импульсного шума является нелинейной операцией, а адаптивная винеровская фильтрация гауссова шума – линейной.

Рассмотрим ещё пример [3, с. 211], [24] – изображение Луны (рис. 18):

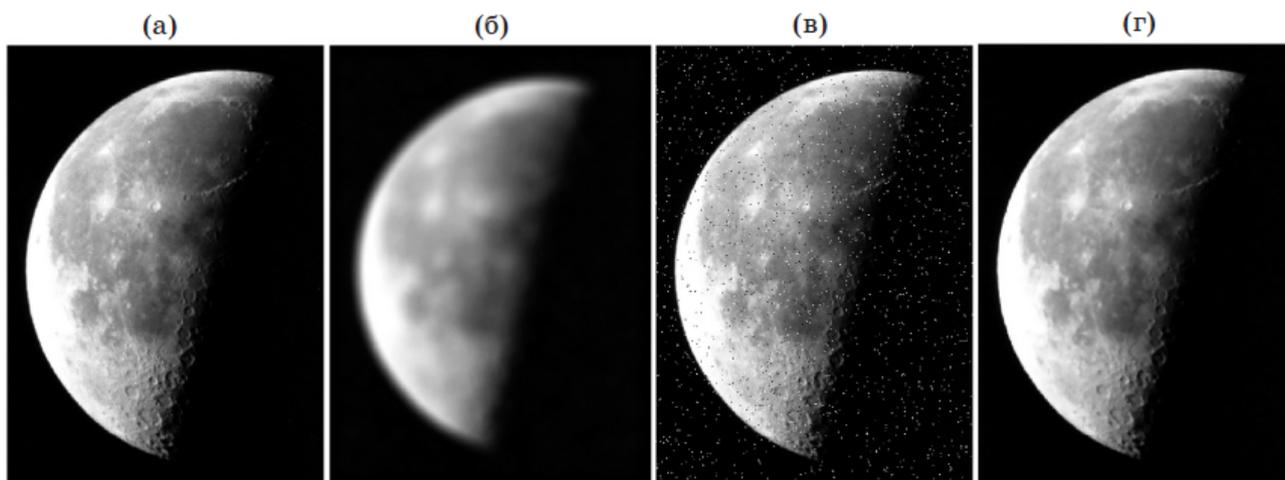


Рис. 18. а – точное изображение Луны; б – изображение, зашумленное импульсным шумом и дефокусированное вместе с шумом; в – изображение (б), восстановленное методом РТ,  $\alpha = 10^{-8}$ ; г – последующая медианная фильтрация (МФ) изображения (в) с маской  $3 \times 3$ ,  $\sigma_{\text{rel}} = 0.0259$

На рис. 19 представлены зависимости  $\sigma_{\text{rel}}(\alpha)$  и  $\sigma_{\text{rel}}(K)$  – относительные погрешности восстановления изображения Луны (см. (40)).

По минимумам этих зависимостей были определены оптимальные значения  $\alpha$  и  $K$ :  $\alpha_{\text{opt}} = 10^{-8}$  и  $\sigma_{\text{rel}}(\alpha) = 0.0259$ ;  $K_{\text{opt}} = 10^{-3.5}$  и  $\sigma_{\text{rel}}(K) = 0.0691$ . Видим, что в данном примере погрешность восстановления изображения методом регуляризации Тихонова (РТ) значительно меньше (в 2.7 раза), чем методом параметрической фильтрации Винера (ПФВ).

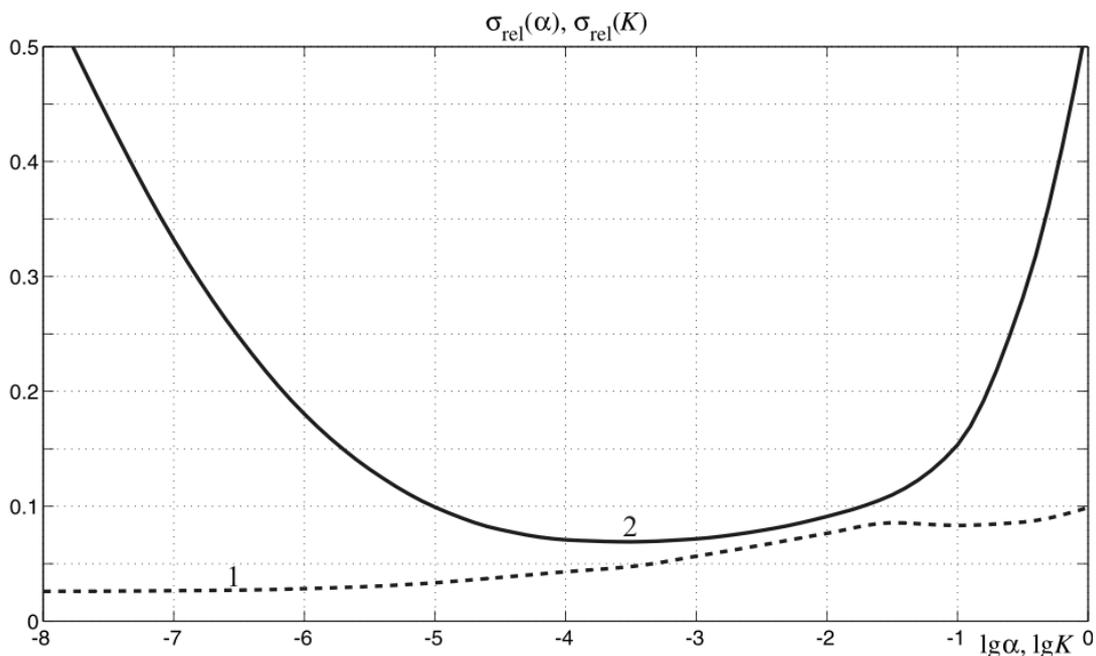


Рис. 19. Относительные погрешности  $\sigma_{rel}(\alpha)$  и  $\sigma_{rel}(K)$  восстановления изображения Луны. 1 – погрешность  $\sigma_{rel}(\alpha)$  методом РТ с последующей МФ; 2 – погрешность  $\sigma_{rel}(\alpha)$  методом ПФВ с последующей МФ

**Влияние неточности параметров искажений на точность восстановления изображений.** Важным является вопрос о точном знании функции рассеяния точки (ФРТ, PSF), а именно, о величине  $\Delta$  и угле  $\theta$  смаза в случае смазывания изображения, а также о величине радиуса  $\rho$  пятна дефокусирования изображения. Дело в том, что методы регуляризации Тихонова (РТ), параметрической фильтрации Винера (ПФВ) и др. чувствительны к погрешностям параметров искажения  $\Delta$ ,  $\theta$  и  $\rho$ .

В качестве иллюстрации на рис. 20а приведено изображение football.jpg  $256 \times 320$ , смазанное под углом  $\theta = 30^\circ$ , величина смаза  $\Delta = 15$  пкс.

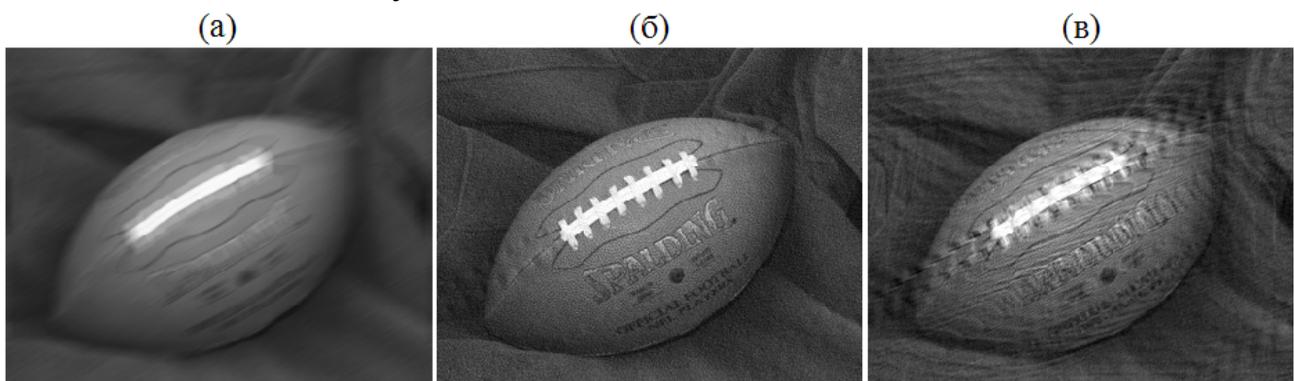


Рис. 20. а – изображение, смазанное под углом  $\theta = 30^\circ$ , смаз  $\Delta = 15$  пкс; б – изображение, восстановленное при точных параметрах  $\theta$  и  $\Delta$  ( $K = 10^{-4}$ ); в – изображение, восстановленное при ошибочных  $\tilde{\theta} = 32^\circ$ ,  $\tilde{\Delta} = 15$  ( $K = 10^{-3}$ )

На рис. 20б представлен результат восстановления изображения путем ре-

шения уравнения (17) методом параметрической фильтрации Винера при точных параметрах смаза  $\theta = 30^\circ$  и  $\Delta = 15$ . А на рис. 20в приведено восстановленное изображение при ошибочных значениях параметров смаза:  $\tilde{\theta} = 32^\circ$  и  $\tilde{\Delta} = 15$ . Хотя  $\tilde{\theta}$  и  $\tilde{\Delta}$  незначительно отличаются от точных  $\theta$  и  $\Delta$ , восстановление получилось неудовлетворительным (рис. 20в).

На рис. 21 представлен аналогичный пример с дефокусированием изображения.

Эти и другие примеры [1, 3, 5, 19, 20, 23] показывают, что методы РТ и ПФВ в совокупности с методами фильтрации шумов МФ и АФВ устойчивы по отношению к шумам  $\delta g$  в измеренном искаженном изображении  $g$  или, говоря математически, по отношению к погрешностям правой части интегрального уравнения (14), (17) или (37).

Однако эти методы неустойчивы также по отношению к неточному знанию параметров искажений  $\Delta$ ,  $\theta$  и  $\rho$  (рис. 20в и 21в), т.е. к погрешностям функции рассеяния точки или, говоря математически, по отношению к погрешностям ядра интегрального уравнения. Если же к методам РТ и ПФВ добавить для совокупности спектральный способ [19, 20], то, как показано в [1, 19, 20], восстановление изображений станет устойчивым (рис. 20б и 21б).

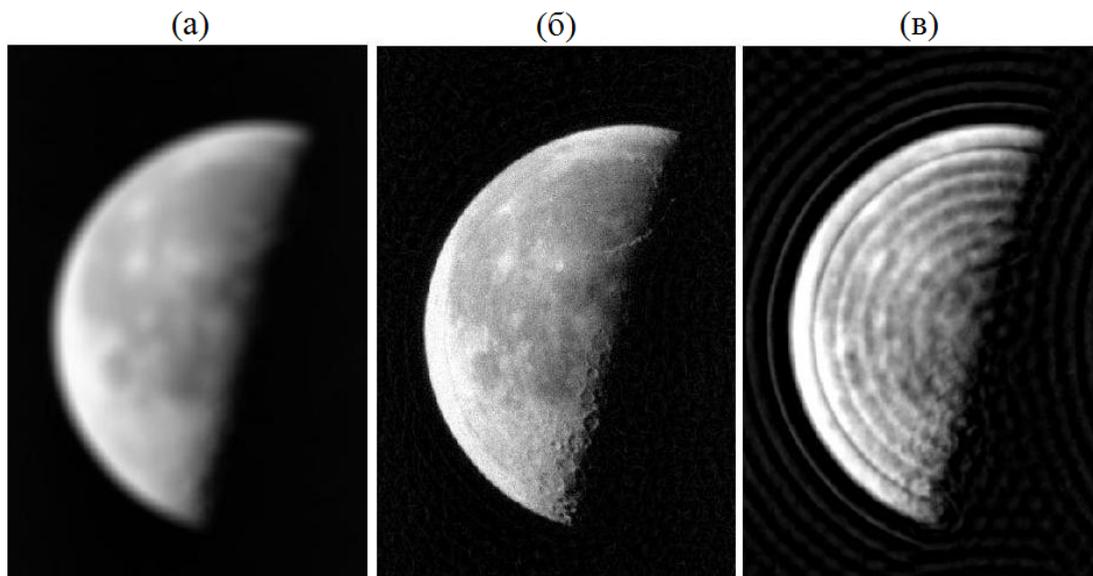


Рис. 21. а – дефокусированное изображение moon.tif 537×358 (ФРТ – однородный диск радиуса  $\rho = 10$  пкс);

б – изображение, восстановленное при точном значении  $\rho$  ( $K = 10^{-4}$ );

в – изображение, восстановленное при ошибочном  $\tilde{\rho} = 12$  ( $K = 10^{-3}$ )

В [25] предложен устойчивый регуляризирующий алгоритм восстановления изображений при неточно известной ФРТ, причем в [25] рассмотрен случай случайных ошибок, а мы рассматриваем случай детерминированных ошибок.

## 4. Обработка спектров

Рассмотрим прямые и обратные задачи спектроскопии – моделирование и восстановление искаженных (сглаженных аппаратной функцией, взаимно перекрывающихся, зашумленных) спектров. Будем использовать математический и программный аппарат почти такой же, как в обработке изображений, поэтому мы объединили в одном учебном пособии обработку изображений и спектров.

### 4.1. Введение в спектральный анализ

**Спектральный анализ и приборы.** Для качественного и количественного исследования веществ широко используется *спектральный анализ* [1–3, 26, 27]. Он заключается [28, с. 708] в определении состава вещества (а также его температуры, плотности, давления и т.д.) на основе изучения спектров излучения (эмиссии), поглощения (абсорбции) и др. [28, с. 303, 354, 713], [29].

Спектральный анализ можно разделить на *широкополосный* (изучение спектра в широкой полосе частот) и *узкополосный* (изучение тонкой структуры одной линии), а также атомный, молекулярный, эмиссионный, абсорбционный, качественный, количественный и т.д.

Под *спектром* подразумевается зависимость интенсивности излучения от длины волны или частоты (часто вместо частоты  $\nu = c/\lambda$  используют  $k = 1/\lambda$  – волновое число [28, с. 85]).

Для разложения излучения в спектр и его регистрации используются оптические *спектральные приборы* [27, с. 23], [28, с. 703–708], [30, с. 309–312]. Каждый такой прибор состоит из трех основных частей: осветительной (необязательная часть), спектральной и регистрирующей. В зависимости от способа регистрации спектра различают следующие *приборы*:

- *спектроскопы* (с визуальной регистрацией);
- *спектрометры* и др. [30, с. 127–129].

Спектр может быть получен следующими путями: по разложению луча света при его прохождении через призму (в призмных спектроскопах), путем вычисления преобразования Фурье от двух интерферирующих лучей (в Фурье-спектрометрах) и т.д.

На рис. 22 представлен принцип работы призмного спектроскопа.

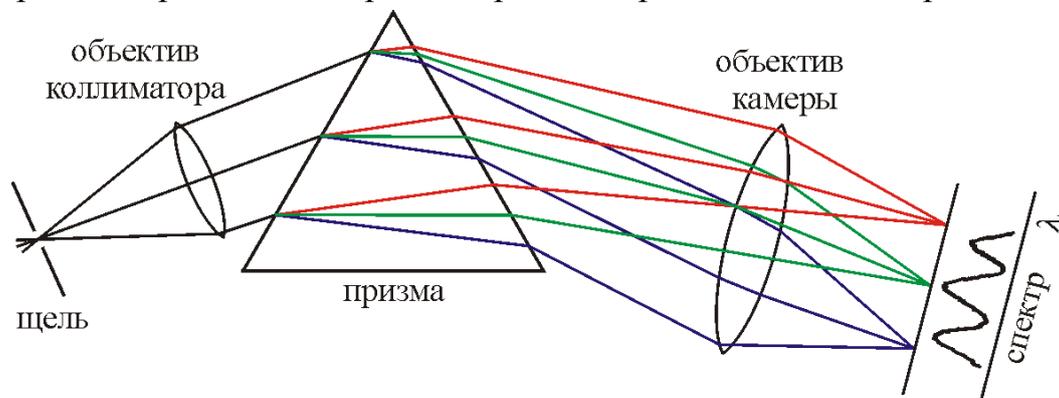


Рис. 22. Принцип работы призмного спектроскопа

Лучи света, содержащие компоненты с различными длинами волн, проходят через щель, через объектив (линзу) коллиматора (коллиматор – устройство для получения параллельных лучей), а также через призму и испытывают в призме два преломления (при входе в призму и при выходе из неё).

При этом коэффициент преломления зависит от длины волны  $\lambda$  (чем больше  $\lambda$ , тем меньше преломление), поэтому при прохождении через призму каждый луч разлагается на ряд лучей с различными  $\lambda$  и в результате (после прохождения лучей через фокусирующий объектив камеры) на выходе получается *спектр* – зависимость интенсивности света от длины волны  $\lambda$ .

На рис. 23 представлен принцип работы Фурье-спектрометра.

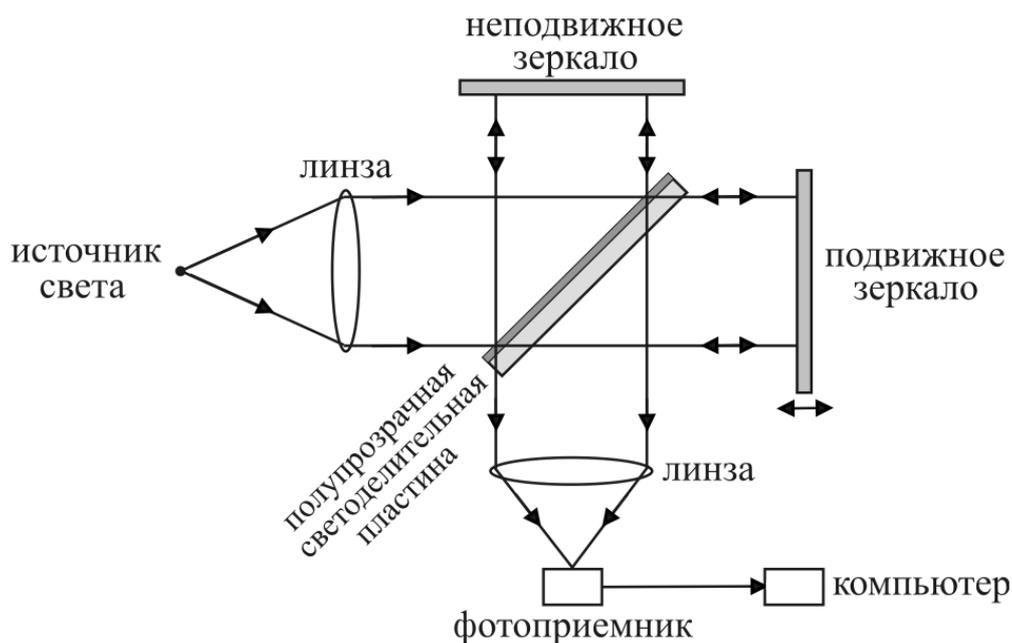


Рис. 23. Принцип работы оптического Фурье-спектрометра

В Фурье-спектрометре призма не используется. Свет от источника после прохождения через линзу коллиматора разделяется на два пучка с помощью полупрозрачной светоделительной пластины. Затем один из пучков отражается от неподвижного зеркала, а другой – от подвижного зеркала. Перемещение подвижного зеркала позволяет изменять разность хода лучей пучков. Далее оба пучка соединяются в фотоприемнике, где они интерферируют и получается интерферограмма (интенсивность сигнала в функции разности хода двух пучков). В заключении получение спектра в функции длины волны или частоты производится с помощью преобразования Фурье от интерферограммы в компьютере.

**Типы спектров.** Спектры бывают (см. рис. 24) [28, с. 712], [30, с. 647]:

– *непрерывные*, или *сплошные* (примеры: спектры расплавленного металла, горячего газового потока в сопле ракеты и т.д.) – это спектры от веществ с повышенной плотностью;

– *дискретные*, или *линейчатые*, состоящие из почти монохроматических спектральных линий с дискретными значениями длины волны  $\lambda$  или частоты  $\nu$  (примеры: спектры веществ, находящихся в глубоком вакууме, в частности, спектры рассеянных межзвездных туманностей);

– *полосатые*, состоящие из ряда полос, каждая из которых, в свою очередь, состоит из набора близких дискретных линий (пример: спектры веществ, находящихся в неглубоком вакууме, в частности, спектры люминесцентных ламп и ламп «дневного света»);

– *сложные*, или *комбинированные*, состоящие, например, из непрерывного спектра и ряда дискретных линий.

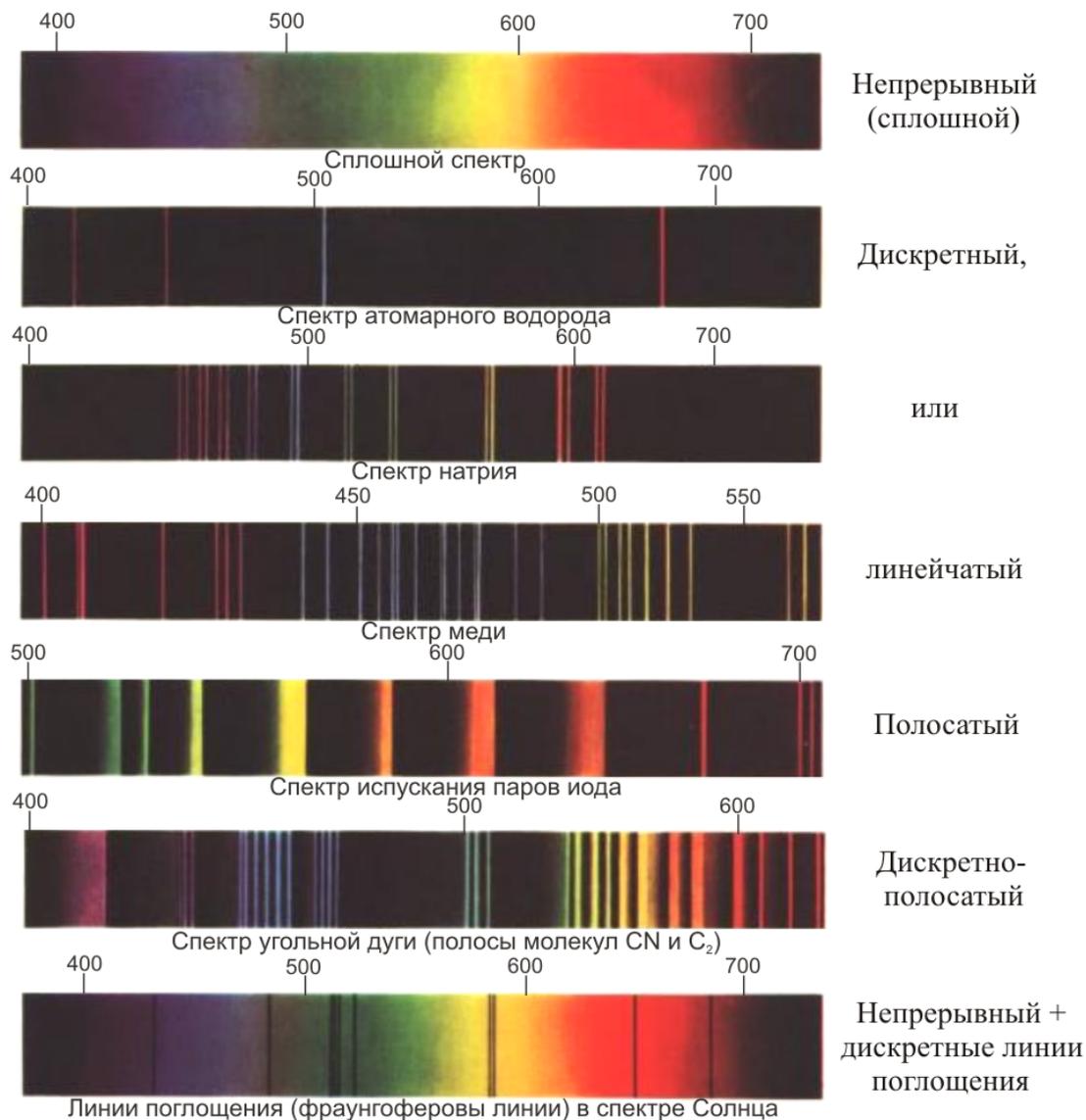


Рис. 24. Типы спектров (указаны длины волн в нм)

**Области применения спектрального анализа:**

- *физика* (изучение спектров газов, жидкостей, металлов, плазмы);
- *астрофизика* (изучение спектров планет, звезд, галактик, комет);
- *металлургия* (определение по спектру состояния расплавленного металла, например, в домне);
- *химия и ЯМР-спектроскопия* (определение по ЯМР-спектру химического состава вещества);
- *геофизика* (разведка руд, минералов, нефти, газа) и т.д.

## 4.2. Восстановление непрерывных спектров

Перейдем к математическому описанию прямой и обратной задач спектроскопии в случае, когда спектр является непрерывным (сплошным) [1–3, 26–28, 31, 32]. Введем следующие важные понятия: аппаратная функция (АФ) спектрометра, истинный и измеренный спектры и др.

### Аппаратная функция спектрального прибора.

Определение. *Аппаратной функцией* (спектральной чувствительностью, инструментальным контуром, функцией пропускания, функцией щели)  $K(\lambda, \lambda')$  будем называть измеренную спектрометром интенсивность дискретной линии единичной интенсивности и частоты  $\lambda'$  при настройке спектрометра на частоту  $\lambda$  (см. рис. 25).

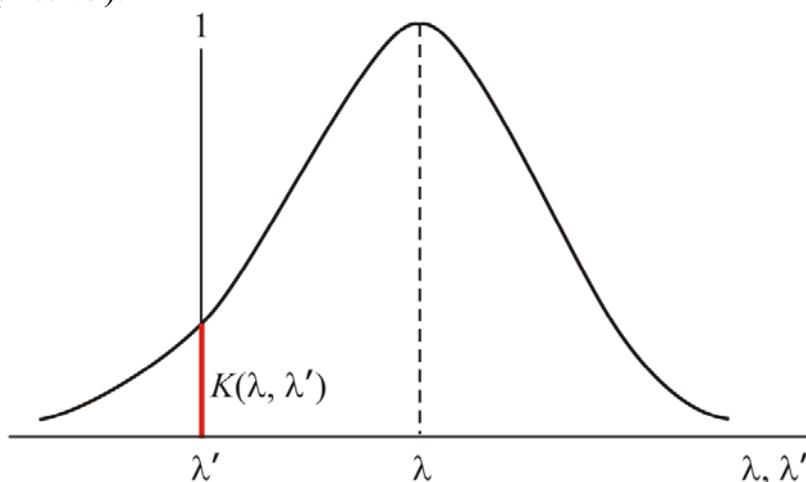


Рис. 25. Аппаратная функция (АФ) спектрометра  $K(\lambda, \lambda')$

Форма АФ может быть различной, например, гауссианой в зависимости от типа щели, дифракции, аберраций и т.д. [26] (см. дальше). Кроме того, в узкополосной спектроскопии форма АФ практически одинакова при различных  $\lambda$ , изменяющихся в узких пределах, и АФ называется разностной (частотноинвариантной):  $K = K(\lambda - \lambda')$  и может записываться как функция одной переменной:  $K = K(\lambda)$ . В случае же широкополосной спектроскопии (когда исследуется спектр, например, от фиолетового до красного цветов) форма АФ изменяется с изменением  $\lambda$  (становится шире с ростом  $\lambda$ ).

**Истинный и измеренный спектры.** Обозначим через  $z(\lambda)$  *истинный спектр* – зависимость интенсивности излучения от длины волны  $\lambda$ , неискаженная мешающими факторами (конечной шириной АФ спектрометра, шумами и т.д.), а через  $\tilde{y}(\lambda)$  – *измеренный* (экспериментальный) *спектр* – это спектр, полученный в результате сканирования по длине волны  $\lambda$  (в некотором диапазоне длин волн) реальным спектрометром, а именно, с учетом его АФ  $K(\lambda, \lambda')$  и при наличии различных шумов (погрешностей измерений, турбулентностей в среде, через которую идет излучение, и т.д.). С учетом этих факторов, истинный спектр  $z(\lambda)$  – это спектр, который был бы измерен при бесконечно узкой (идеальной) АФ и в отсутствие шумов.

Измеренный спектрометром спектр  $\tilde{u}(\lambda)$  обычно более сглажен, чем  $z(\lambda)$  (в  $\tilde{u}(\lambda)$  неразрешены близкие линии, заглажена тонкая структура некоторых линий) и  $\tilde{u}(\lambda)$  обычно зашумлен (в результате слабые линии «тонут» в шуме).

Рис. 26 иллюстрирует сказанное (АФ неразностная).

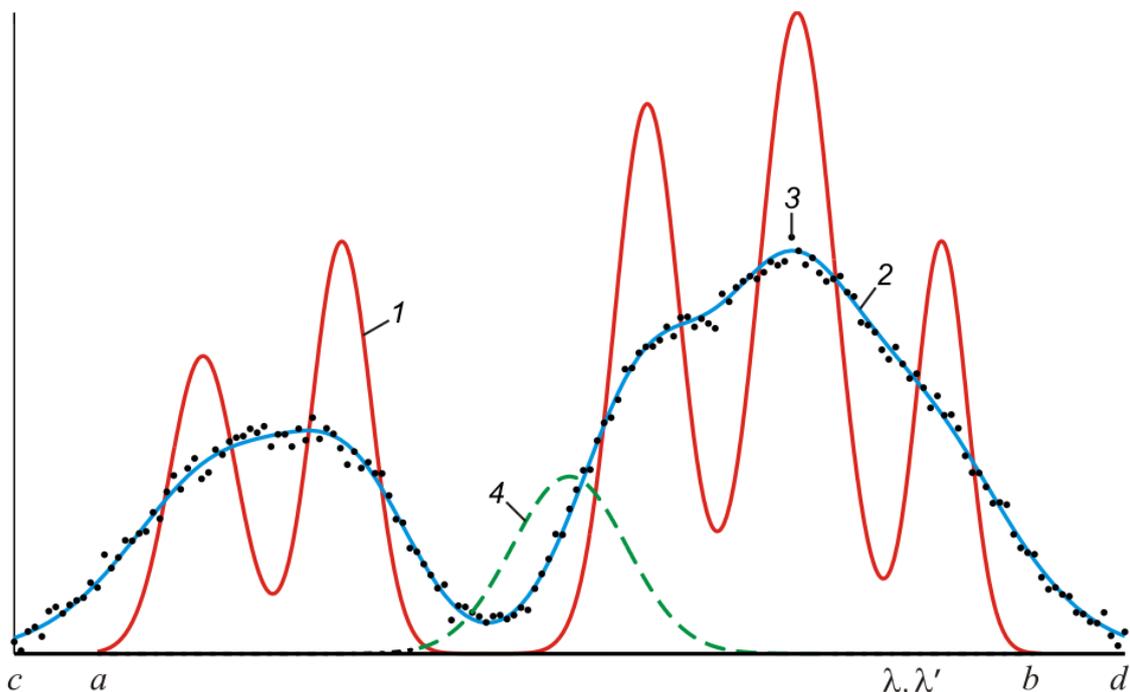


Рис. 26. 1 – истинный спектр  $z(\lambda)$ ; 2 – измеренный спектр  $u(\lambda)$ , сглаженный АФ, но без шума; 3 – измеренный с шумом спектр  $\tilde{u}(\lambda)$ ; 4 – сечение АФ  $K(\lambda, \lambda')$  в функции  $\lambda'$  при некотором фиксированном  $\lambda$

Рис. 26 показывает: в истинном спектре  $z(\lambda)$  5 линий, но в измеренном спектре  $\tilde{u}(\lambda)$  они не разрешились из-за широкой АФ, а также зашумлены.

Возникает задача: по экспериментальному спектру  $\tilde{u}(\lambda)$  и АФ  $K(\lambda, \lambda')$  путем математико-компьютерной обработки восстановить истинный спектр  $z(\lambda)$ . Успешное решение этой задачи позволит повысить разрешающую способность спектрометра, а значит, качество спектрального анализа.

Задача определения истинного спектра  $z(\lambda)$  на основе экспериментального спектра  $\tilde{u}(\lambda)$  и АФ  $K(\lambda, \lambda')$  называется *задачей редукции* (приведения) *профиля спектра к идеальному спектральному прибору* (другое название: *исключение АФ*) [10, 26].

Чтобы повысить разрешающую способность спектрометра, можно использовать (ср. [27, с. 27]) физико-техничко-коммерческий путь, а именно, использовать более совершенный (а, значит, более дорогой) спектрометр. Но можно использовать и математико-компьютерный путь, а именно, выполнить математическую обработку (с помощью компьютера) результатов измерений путем решения соответствующего уравнения. Это – один из вариантов известной *редукционной проблемы Рэлея* [32].

**Прямая и обратная задачи спектроскопии.** Опишем математически *прямую задачу* (моделирование экспериментального искаженного спектра). Измеренное значение интенсивности  $\tilde{u}(\lambda)$  при настройке спектрометра на длину волны  $\lambda$  равно интегралу (сумме) по всем интенсивностям  $z(\lambda')$  с весовой функцией, равной  $K$ , т.е. (ср. [27, с. 25])

$$\tilde{u}(\lambda) = \int_a^b z(\lambda') K(\lambda, \lambda') d\lambda' + \delta u, \quad c \leq \lambda \leq d. \quad (41)$$

Математически в (41) задаются истинный спектр  $z(\lambda)$  (кривая 1 на рис. 26) и АФ  $K(\lambda, \lambda')$  (кривая 4 на рис. 26), а рассчитывается экспериментальный (сглаженный) спектр  $u(\lambda)$  (кривая 2 на рис. 26) и добавляется погрешность  $\delta u$  (3 на рис. 26).

Теперь опишем математически *обратную задачу* (восстановление истинного спектра  $z(\lambda)$ ). Поменяем местами левую и правую части в (41), получим:

$$\int_a^b K(\lambda, \lambda') z(\lambda') d\lambda' = \tilde{u}(\lambda), \quad c \leq \lambda \leq d. \quad (42)$$

В соотношении (42) известны (измерены или заданы)  $\tilde{u}(\lambda)$ ,  $K(\lambda, \lambda')$ ,  $a$ ,  $b$ ,  $c$  и  $d$ , а  $z(\lambda')$  является искомой функцией. Соотношение (42) есть *интегральное уравнение* (ИУ) Фредгольма I рода относительно  $z(\lambda')$ .

Математические (и физические) названия входящих в (42) функций:  $K(\lambda, \lambda')$  – ядро (АФ),  $z(\lambda')$  – искомая функция (истинный спектр),  $\tilde{u}(\lambda)$  – правая часть (измеренный спектр).

Если  $K(\lambda, \lambda') = K(\lambda - \lambda')$ , то (42) записывается в виде:

$$\int_0^{\infty} K(\lambda - \lambda') z(\lambda') d\lambda' = \tilde{u}(\lambda), \quad 0 \leq \lambda < \infty. \quad (43)$$

Соотношение (43) есть ИУ Фредгольма I рода типа свертки на полуоси.

Если пределы  $[a, b]$  и  $[c, d]$  широкие, то мы имеем дело с широкополосной спектроскопией и в этом случае нужно учитывать изменение формы  $K(\lambda, \lambda')$  с изменением  $\lambda$ . Если же пределы  $[a, b]$  и  $[c, d]$  узкие (например, при изучении тонкой структуры одной линии), то это будет задача обработки спектра в рамках узкополосной спектроскопии и следует использовать ИУ вида:

$$\int_a^b K(\lambda - \lambda') z(\lambda') d\lambda' = \tilde{u}(\lambda), \quad c \leq \lambda \leq d. \quad (44)$$

Задача решения ИУ (42)–(44) является *некорректной* (существенно неустойчивой). На рис. 27 приведен результат численного решения ИУ (42) методом квадратур без дополнительного использования устойчивого метода. Решение получилось чрезвычайно неустойчивым – в виде так называемой «пилы» большой амплитуды, ничего общего не имеющей с точным решением  $\bar{z}(\lambda)$ .

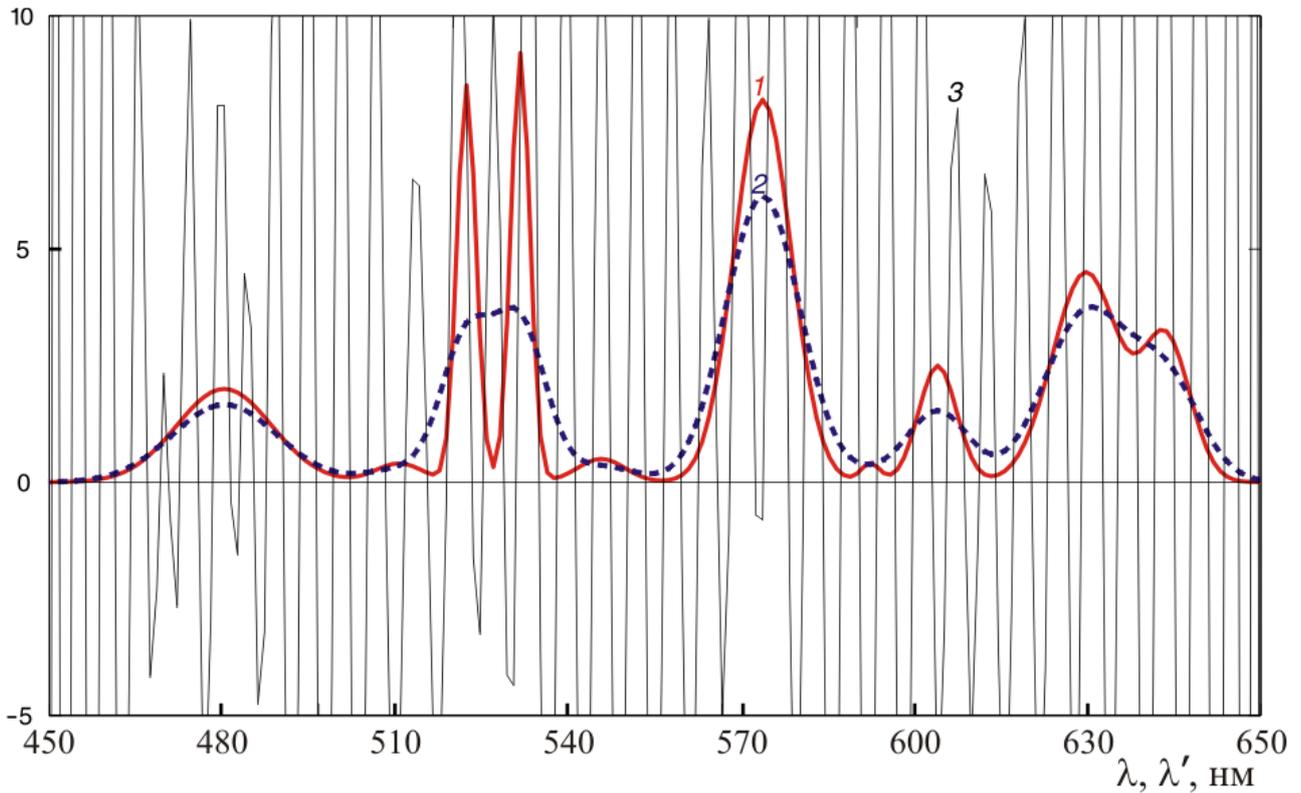


Рис. 27. 1 – точный спектр  $\bar{z}(\lambda)$ ; 2 – сглаженный (измеренный) спектр  $\tilde{u}(\lambda)$ ; 3 – неустойчивое (пилообразное) решение  $z(\lambda)$

**Метод регуляризации Тихонова.** В п. 3.2 изложены: метод ПФ и регуляризации Тихонова (согласно (25), (26)), а также метод параметрической фильтрации Винера (согласно (27)) решения 1-мерного ИУ Фредгольма I рода типа свертки (14), напоминающего ИУ (43) и (44). Эти методы доведены до m-функций `deconvreg.m`, `deconvwnr.m` и `desmearingf.m`. Однако данные m-функции ориентированы на задачу устранения смазывания изображений и практически не подходят для решения обратной задачи спектроскопии.

Кроме того, в п. 3.2 не был рассмотрен метод регуляризации Тихонова с квадратурами для решения 1-мерного ИУ Фредгольма I рода общего типа (42).

Поэтому приведем вариант решения ИУ Фредгольма I рода методом квадратур с регуляризацией Тихонова, ориентированный на обратную задачу спектроскопии (а также другие задачи).

Будем иметь в виду ИУ Фредгольма I рода общего типа (42), но запишем его в иных обозначениях (ср. [1–3]):

$$Ay \equiv \int_a^b K(x,s)y(s)ds = f(x), \quad c \leq x \leq d, \quad (45)$$

где  $K(x,s)$  – ядро,  $y(s)$  – искомая функция,  $f(x)$  – правая часть,  $A$  – оператор.

Пусть правая часть  $f(x)$  задана на неравномерной  $x$ -сетке узлов:

$$c = x_1 < x_2 < \dots < x_i < \dots < x_m = d, \quad (46)$$

а решение  $y(s)$  ищется на другой также неравномерной  $s$ -сетке узлов:

$$a = s_1 < s_2 < \dots < s_j < \dots < s_n = b. \quad (47)$$

Заменим интеграл в (45) при каждом  $x$  конечной суммой по формуле трапеций, получим систему  $m$  линейных алгебраических уравнений (СЛАУ) относительно  $n$  неизвестных  $y_j \equiv y(s_j)$ :

$$\sum_{j=1}^n r_j K_{ij} y_j = f_i, \quad i = \overline{1, m}, \quad (48)$$

где

$$r_1 = \frac{s_2 - s_1}{2}, \quad r_j = \frac{s_{j+1} - s_{j-1}}{2}, \quad j = \overline{2, n-1}, \quad r_n = \frac{s_n - s_{n-1}}{2} \quad (49)$$

– коэффициенты квадратурной формулы трапеций. Запишем СЛАУ (48) в виде:

$$Ay = f, \quad (50)$$

где  $a_{ij} = r_j K_{ij}$ ,  $i = \overline{1, m}$ ,  $j = \overline{1, n}$  – элементы матрицы  $A_{m \times n}$ ;  $y_{n \times 1}$  – искомый вектор-столбец;  $f_{m \times 1}$  – правая часть (вектор-столбец).

В методе регуляризации 0-го порядка Тихонова вместо СЛАУ (50) решается СЛАУ [2, с. 193, формула (8.6)]:

$$(\alpha E + A^T A) y_\alpha = A^T f, \quad (51)$$

где  $\alpha > 0$  – параметр регуляризации;  $E$  – единичная матрица (диагональные элементы – единицы, остальные – нули);  $A^T$  – транспонированная матрица.

Регуляризованное решение  $y_\alpha$  можно записать в виде:

$$y_\alpha = (\alpha E + A^T A)^{-1} A^T f. \quad (52)$$

Разработана m-функция Tikh0.m для решения ИУ (45) методом квадратур с регуляризацией Тихонова 0-го порядка согласно (46)–(52):

```
%% Tikh0.m, Сизиков, m-функция, MatLab.
%Решение интегрального уравнения Фредгольма I рода
%методом квадратур и регуляризации 0-го порядка Тихонова
%(а также методом квадратур и МНК Гаусса при alpha=0).
%Входные параметры (аргументы):
%f - правая часть f(x) - вектор-строка, 1xm double,
%x - сетка узлов по x (необязательно равномерная), 1xm double,
%s - сетка узлов по s (необязательно равномерная), 1xn double,
%kern - ядро K(x,s) интегрального уравнения (m-функция), double,
%alpha>0 - параметр регуляризации, double.
%Выходные параметры (аргументы):
%y_alpha - регуляризованное решение y_alpha(s), double,
%y0_alpha - регуляризованное решение y_alpha(s)>=0, double.
%Возможные варианты обращений к Tikh0:
%y_alpha = Tikh0(f,x,s,@kern,alpha);
%или
%[y_alpha,y0_alpha] = Tikh0(f,x,s,@kern,alpha);
%Литература: Сизиков В.С. Обратные прикладные задачи и
%MatLab. - СПб.: Лань, 2011, с.206-207.
%Сизиков В.С., Кривых А.В. Оптика и спектроскопия. 2014,
%т.117, № 6, с. 1040-1048, формула (3).
function [y_alpha,y0_alpha] = Tikh0(f,x,s,kern,alpha)
```

```

m = length(x); %длина массива x
n = length(s); %длина массива s
%Формируем вектор-строку квадратурных коэффициентов 1xn:
r = [s(2)-s(1), s(3:n)-s(1:n-2), s(n)-s(n-1)]/2;
%Формируем матрицу A mnx:
A = repmat(r,m,1).*kern(repmat(x',1,n),repmat(s,m,1));
%Решение ИУ в виде решения СЛАУ методом исключения Гаусса:
y_alpha = ((alpha*eye(n)+A'*A)\(A'*f'))'; %рег-ция 0-го порядка
%Корректировка решения (если y<0, то y=0):
if (nargout==2)
    y0_alpha = y_alpha; y0_alpha(y0_alpha<0) = 0; %доп. условие: y>=0
end %if

```

Разработанная m-функция Tikh0.m может быть использована для решения ИУ (42), которое запишем в виде (аналогичном (45)):

$$Az \equiv \int_a^b K(\lambda, \lambda') z(\lambda') d\lambda' = \tilde{u}(\lambda), \quad c \leq \lambda \leq d \quad (53)$$

или в виде операторного уравнения (ср. (50)):

$$Az = u. \quad (54)$$

Полагаем, что вместо точных  $u$  и  $K$  известны  $\tilde{u}$  и  $\tilde{K}$  такие, что  $\|\tilde{u} - u\| \leq \delta$ ,  $\|\tilde{K} - K\| \leq \xi$ , где  $\delta$  и  $\xi$  – погрешности правой части  $u$  и ядра  $K$ .

В методе регуляризации 0-го порядка Тихонова вместо (54) решается операторное уравнение, а при использовании метода квадратур – СЛАУ (ср. (51)):

$$(\alpha E + A^T A) z_\alpha = A^T u. \quad (55)$$

**О способе выбора параметра регуляризации  $\alpha$ .** Важным является вопрос о выборе параметра регуляризации  $\alpha$  и об оценке погрешности регуляризованного решения  $z_\alpha$ . Для выбора  $\alpha$  обычно используются следующие способы: принцип невязки, обобщенный принцип невязки и др.

В данном учебном пособии предлагается использовать для выбора  $\alpha$  способ обучающих примеров-спектров (способ модельных спектров) [1–3, 33], в котором используются обучающие модельные спектры, «близкие» к восстанавливаемому спектру. Этот способ позволяет оценить погрешность восстановления спектра, не зная истинного спектра.

Рассмотрим кратко этот способ [34]. Пусть нам дан измеренный спектр  $u(\lambda)$  (см. рис. 28). Известно, что АФ спектрометра является дисперсионной [26]

$$K(\lambda, \lambda') = \frac{w(\lambda)/2\pi}{(\lambda - \lambda')^2 + [w(\lambda)/2]^2}, \quad (56)$$

где  $w(\lambda) = q\lambda$  – ширина АФ по уровню 0.5, причем  $q = 0.015$ . Назовем данный спектр примером  $P$ . Анализ рис. 28 показывает, что в спектре-примере  $P$  имеется 8–10 линий, но не все они видны.

Поэтому в качестве второго (модельного, обучающего) примера  $Q$  был смоделирован «близкий» к оригиналу  $P$  пример, истинный спектр которого состоит из 8–10 спектральных линий (в виде гауссиан).

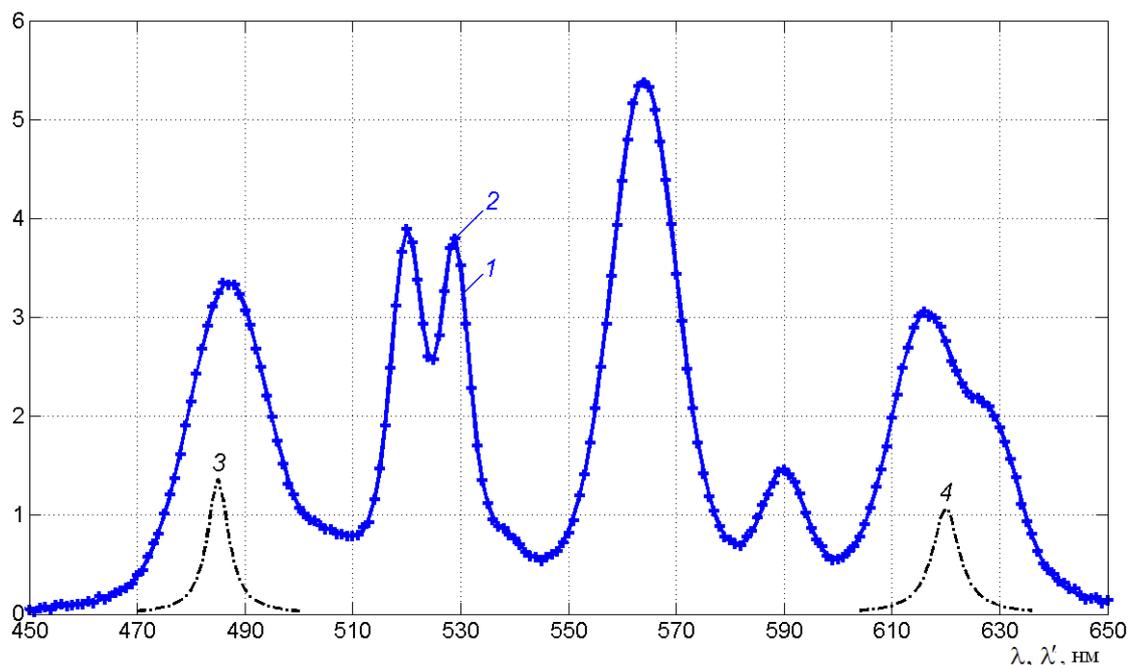


Рис. 28. Пример  $P$ . 1 – точный спектр  $\bar{u}(\lambda)$ ; 2 – зашумленный спектр  $\tilde{u}(\lambda)$  и два сечения АФ: 3 –  $10K(485, \lambda')$  и 4 –  $10K(620, \lambda')$

В ряде работ [1–3, 15, 33, 34] показано, что

$$\sigma_{\text{rel}}(\alpha) \equiv \frac{\|\Delta z_{\alpha}\|}{\|z\|} \leq \varepsilon(\alpha), \quad (57)$$

где

$$\varepsilon(\alpha) = \frac{\|A\|}{2\sqrt{\alpha}} \eta + \frac{\alpha}{\alpha + g}, \quad (58)$$

$\eta = \delta_{\text{rel}} + \xi_{\text{rel}}$ . Здесь  $\sigma_{\text{rel}}(\alpha) = \|\Delta z_{\alpha}\| / \|z\|$  – относительная погрешность восстановления спектра, а  $\varepsilon(\alpha)$  – огибающая погрешности, причем  $g$  определяется из условия  $\sigma_{\text{rel}}(\alpha) = \varepsilon(\alpha)$ . Оценки (57) и (58) для примеров  $P$  и  $Q$  одинаковы  $\sigma_{\text{rel}}(\alpha)$  и  $\varepsilon(\alpha)$ , так как у них одинаковы  $\|A\|\eta$ .

На рис. 29 представлены полученные зависимости  $\sigma_{\text{rel}}(\alpha)_Q$  для ряда «близких» модельных примеров и ряда значений погрешности  $\eta$  (область между кривыми 1 и 2). Заметим, что кривая 3 в способе моделирования полагается неизвестной; она приведена лишь для иллюстрации.

На рис. 29 представлено также несколько огибающих  $\varepsilon(\alpha)$  согласно (58) при  $\|A\| = 0.843$  и  $\eta = \delta_{\text{rel}} + \xi_{\text{rel}} = 2 \cdot 10^{-2}$  для ряда значений  $g$  от 0 до 0.1. Выбрано то значение  $g$ , при котором одна из кривых  $\varepsilon(\alpha)$  касается набора кривых  $\sigma_{\text{rel}}(\alpha)$ , а именно,  $g = 0.045$ . Этому соответствует значение параметра регуляризации  $\alpha = \alpha_g = 10^{-2.2}$ . Из рис. 29 видно, что несмотря на разброс кривых  $\sigma_{\text{rel}}(\alpha)$  и  $\varepsilon(\alpha)$ , значение  $g$  и, как следствие,  $\alpha$  оценивается надежно.

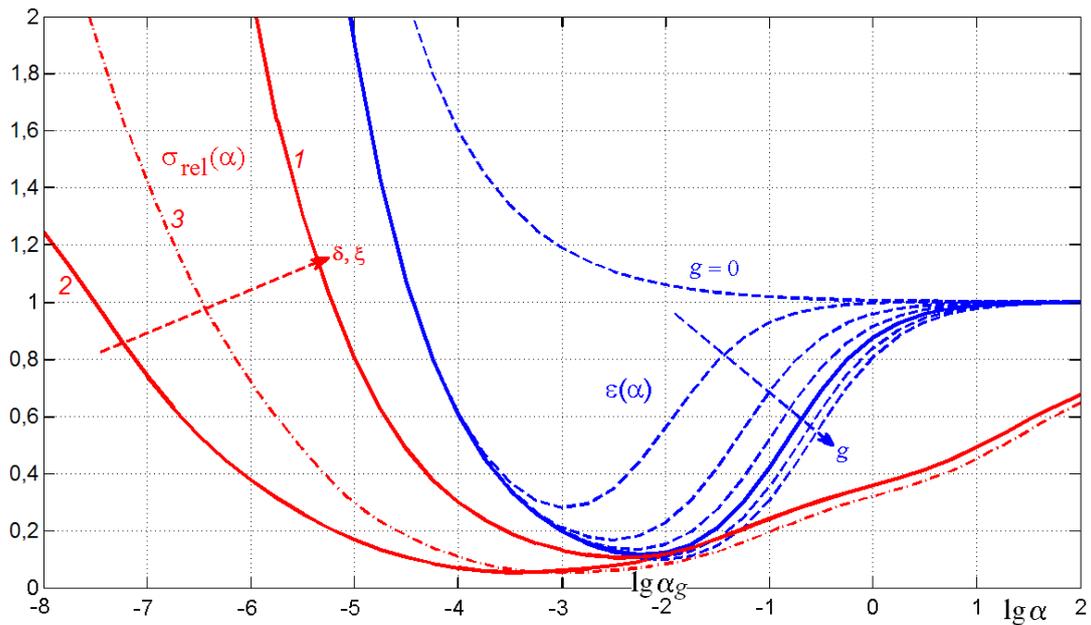


Рис. 29. Относительные погрешности  $\sigma_{\text{rel}}(\alpha)$  для примеров  $Q$  (1 и 2 – ограничения сверху и снизу, а 3 –  $\sigma_{\text{rel}}(\alpha)_P$ ) и огибающие  $\varepsilon(\alpha)$  для ряда значений параметра  $g$

На рис. 30 приведено решение (восстановленный спектр) при  $\alpha_g = 10^{-2.2}$ ,  $\sigma_{\text{rel}}(\alpha_g) = 0.073 = 7.3\%$ . Из рис. (30) видно, насколько точно восстановился спектр: разрешились близкие линии и выделились слабые линии.

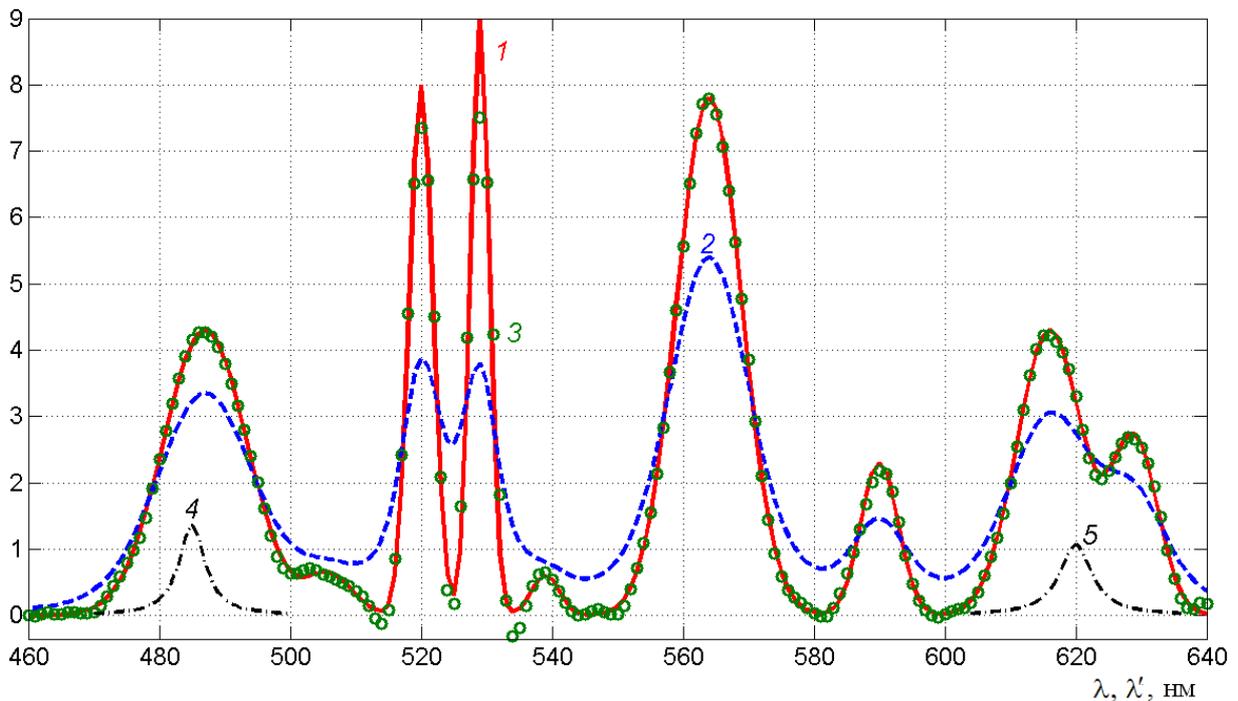


Рис. 30. Пример  $P$ . 1 – истинный спектр  $z_P(\lambda)$ ; 2 – измеренный спектр  $u_P(\lambda)$ ; 3 – восстановленный спектр  $z_{\alpha P}(\lambda)$  при  $\alpha = \alpha_g = 10^{-2.2}$  и два сечения АФ: 4 –  $10 K(485, \lambda')$  и 5 –  $10 K(620, \lambda')$

### 4.3. Разделение перекрывающихся спектральных линий

**Постановка задачи.** Рассмотрим ещё одну актуальную обратную задачу спектроскопии – задачу разделения (сепарации) непрерывных перекрывающихся спектральных линий [35–41]. Проиллюстрируем задачу рисунком 31. На нем отображены  $z_j(\lambda)$ ,  $j = \overline{1, N}$  – профили  $N = 3$  перекрывающихся линий. Однако профили  $z_j(\lambda)$  неизвестны, а известен лишь суммарный измеренный спектр

$$Z(\lambda) = \sum_{j=1}^N z_j(\lambda), \quad a \leq \lambda \leq b, \quad (59)$$

где  $j$  – номер линии,  $[a, b]$  – пределы изменения  $\lambda$ . Требуется на основе суммарного (обычно зашумленного) измеренного спектра  $Z(\lambda)$  определить число линий  $N$  и восстановить профили отдельных линий-компонент  $z_j(\lambda)$ . При этом можно моделировать профили линий некоторыми функциями (гауссианами, лоренцианами и т.д.) и использовать дополнительную информацию о линиях (начальные приближения параметров линий, диапазоны их изменения и т.д.).

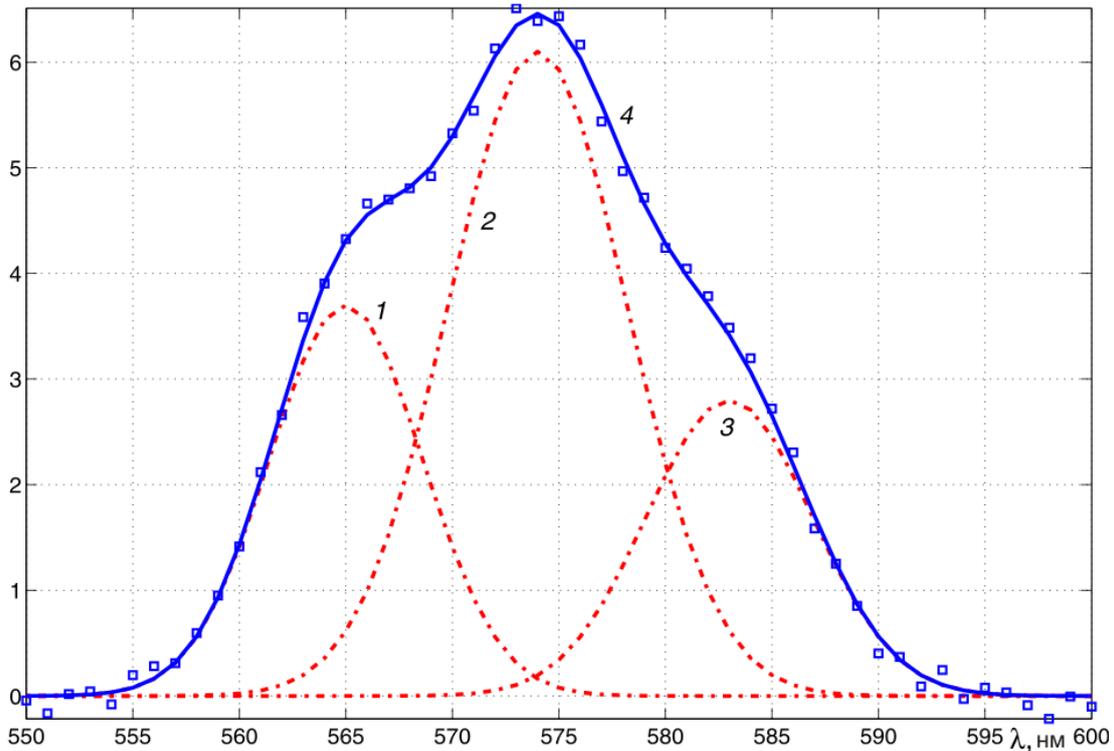


Рис. 31. Исходные спектры. 1, 2, 3 – истинные перекрывающиеся линии (гауссианы)  $z_1(\lambda)$ ,  $z_2(\lambda)$ ,  $z_3(\lambda)$ ; 4 – суммарный спектр  $Z(\lambda) = z_1(\lambda) + z_2(\lambda) + z_3(\lambda)$  незашумленный (непрерывная линия) и зашумленный (квадратики) 5-процентным шумом ( $\delta z = 0.1$ ) [39, рис. 3]

Строго говоря, данная задача является некорректной (может не существовать решения, оно может быть неединственным или неустойчивым) [11–15].

Однако за счет использования дополнительной информации и, например, метода регуляризации Тихонова задача может стать условно корректной [14].

**Моделирование линий и оценка их числа.** Используем моделирование линий гауссианами и лоренцианами. При этом для оценки числа линий  $N$  используем известный *метод производных* суммарного спектра  $d^l Z(\lambda)/d\lambda^l$ , где обычно  $l = \overline{1,4}$ , т.е. используются производные от 1-го до 4-го порядков [35, 37, 38, 41].

**Моделирование линий гауссианами.** В работе [35] впервые предложен *метод производных* с использованием производной лишь 1-го порядка, а в последующих работах [37, 38, 41] и др. использованы производные вплоть до 4-го порядка. Моделируем каждую линию  $z_j(\lambda)$  *гауссианой* (гауссовой функцией):

$$z_j(\lambda) = A_j \exp\left(-\frac{(\lambda - \bar{\lambda}_j)^2}{2\sigma_j^2}\right), \quad j = \overline{1, N}, \quad (60)$$

где  $A_j$  – амплитуда линии,  $\bar{\lambda}_j$  – координата максимума,  $\sigma_j^2$  – дисперсия, при этом  $\tau_j = \sqrt{2 \ln 2} \sigma_j = 1.1773 \sigma_j$  – полуширина гауссианы по уровню 0.5 [33, 42].

Производные порядков  $l = 1, 2, 3, 4$  от  $z_j(\lambda)$  по  $\lambda$  равны (ср. [38–41, 43])

$$z'_j(\lambda) = -\frac{A_j}{\sigma_j} \exp\left(-\frac{(\lambda - \bar{\lambda}_j)^2}{2\sigma_j^2}\right) \cdot \left(\frac{\lambda - \bar{\lambda}_j}{\sigma_j}\right), \quad (61)$$

$$z''_j(\lambda) = -\frac{A_j}{\sigma_j^2} \exp\left(-\frac{(\lambda - \bar{\lambda}_j)^2}{2\sigma_j^2}\right) \cdot \left(-\frac{(\lambda - \bar{\lambda}_j)^2}{\sigma_j^2} + 1\right), \quad (62)$$

$$z'''_j(\lambda) = \frac{A_j}{\sigma_j^3} \exp\left(-\frac{(\lambda - \bar{\lambda}_j)^2}{2\sigma_j^2}\right) \cdot \left(-\frac{(\lambda - \bar{\lambda}_j)^2}{\sigma_j^2} + 3\right) \cdot \left(\frac{\lambda - \bar{\lambda}_j}{\sigma_j}\right), \quad (63)$$

$$z^{IV}_j(\lambda) = \frac{A_j}{\sigma_j^4} \exp\left(-\frac{(\lambda - \bar{\lambda}_j)^2}{2\sigma_j^2}\right) \cdot \left(\frac{(\lambda - \bar{\lambda}_j)^4}{\sigma_j^4} - 6\frac{(\lambda - \bar{\lambda}_j)^2}{\sigma_j^2} + 3\right). \quad (64)$$

Из формул (61)–(64) следует, что

$$z'_j(\bar{\lambda}_j) = 0, \quad z''_j(\bar{\lambda}_j) = -\frac{A_j}{\sigma_j^2} < 0, \quad z'''_j(\bar{\lambda}_j) = 0, \quad z^{IV}_j(\bar{\lambda}_j) = 3\frac{A_j}{\sigma_j^4} > 0, \quad (65)$$

т.е. отрицательный экстремум 2-й производной и положительный экстремум 4-й производной соответствуют максимуму  $j$ -й линии, а нулевые значения 1-й и 3-й производных (при  $z''_j(\bar{\lambda}_j) < 0$ ,  $z^{IV}_j(\bar{\lambda}_j) > 0$ ) также указывают на  $j$ -ю линию.

Из выражений для  $z''_j(\bar{\lambda}_j)$  и  $z^{IV}_j(\bar{\lambda}_j)$  в (65) можно определить  $A_j$  и  $\sigma_j$ , решая систему двух уравнений:

$$\left. \begin{aligned} \frac{A_j}{\sigma_j^2} &= -z_j''(\bar{\lambda}_j), \\ \frac{A_j}{\sigma_j^4} &= \frac{1}{3}z_j^{IV}(\bar{\lambda}_j), \end{aligned} \right\} \quad (66)$$

откуда

$$A_j = 3 \frac{[z_j''(\bar{\lambda}_j)]^2}{z_j^{IV}(\bar{\lambda}_j)}, \quad \sigma_j = \sqrt{3 \frac{-z_j''(\bar{\lambda}_j)}{z_j^{IV}(\bar{\lambda}_j)}}, \quad j = \overline{1, N}. \quad (67)$$

Однако реальный (не модельный) измеренный спектр обычно зашумлен и его численное дифференцирование (не по аналитическим формулам (61)–(65)) выполняется со значительными погрешностями (см. [39, 41]). Можно отфильтровать шумы, например, фильтром Савицкого–Голея с помощью *m*-функции `sgolayfilt.m` в системе MatLab [6, с. 193]. Можно также аппроксимировать спектр *сглаживающим сплайном* [2, 12] и дифференцировать сплайн, как это успешно сделано в [1, с. 323].

Кроме того, на практике невозможно дифференцировать непосредственно линии  $z_j(\lambda)$ , а можно дифференцировать только суммарный спектр  $Z(\lambda)$ . В результате параметры отдельных линий будут определяться по производным суммарного спектра с погрешностями, а ряд линий может вообще не проявиться по некоторым производным согласно критериям (65). Поэтому нужно использовать все производные 1–4-го порядков, а также использовать более точные методы сепарации линий (модифицированный метод координатного спуска [39, 41] и др.), полагая в качестве начальных приближений значения  $N$ ,  $A_j$ ,  $\bar{\lambda}_j$  и  $\sigma_j$ , оцененные по методу производных.

В работе [41] выведены также формулы для производных 1–4-го порядков, когда линии аппроксимируются лоренцианами.

**Степень наложения линий.** В работе [38] введён эффективный параметр, характеризующий ситуацию с наложением двух пиков, – *степень наложения пиков* (в наших обозначениях):

$$D = \frac{\tau}{\Delta}, \quad (68)$$

где  $\tau$  – полуширина пиков-линий по уровню 0.5, а  $\Delta$  – расстояние между пиками. Однако формула (68) применима лишь для случая, когда полуширины обоих пиков одинаковы. А для случая, вообще говоря, разных полуширин предлагаем следующую формулу:

$$D = \frac{\bar{\tau}}{\Delta}, \quad (69)$$

где  $\bar{\tau} = (\tau_1 + \tau_2)/2$  – средняя полуширина 1-го и 2-го пиков. Однако формула (69) не учитывает возможного различия амплитуд  $A$  пиков и случая, когда число пиков  $N$  больше двух. Предлагается следующая формула для случая, вообще говоря, разных амплитуд  $A_j$ ,  $j = \overline{1, N}$ , пиков и  $N \geq 2$ :

$$D_{j,j+1} = \frac{\bar{\tau}_{j,j+1}}{\Delta_{j,j+1}} \left( 1 + \frac{|A_j - A_{j+1}|}{A_j + A_{j+1}} \right), \quad j = \overline{1, N-1}, \quad (70)$$

где  $\bar{\tau}_{j,j+1} = (\tau_j + \tau_{j+1})/2$  – средняя полуширина соседних ( $j$  и  $j+1$ ) пиков, а  $\Delta_{j,j+1} = \bar{\lambda}_{j+1} - \bar{\lambda}_j$  – расстояние между соседними пиками.

Сделаем некоторый анализ формул (68)–(70). Если амплитуды линий одинаковы:  $A_j = A_{j+1}$ , то формула (70) переходит в формулу  $D_{j,j+1} = \bar{\tau}_{j,j+1}/\Delta_{j,j+1}$ , а при  $N = 2$  в формулу (69). А если  $A_{j+1} \ll A_j$ , то  $D_{j,j+1} = 2\bar{\tau}_{j,j+1}/\Delta_{j,j+1}$ . Это говорит о том (и формула (70) это подтверждает), что чем шире пики, а также чем они ближе друг к другу и чем больше отличаются их амплитуды, тем больше степень наложения пиков  $D$ , а значит, тем сложнее разделить пики при решении обратной задачи (см. далее).

Будем использовать также следующий параметр:

$$\bar{D} = \frac{1}{N-1} \sum_{j=1}^{N-1} D_{j,j+1} \quad (71)$$

– среднее значение степени наложения всех  $N$  пиков-линий.

**Сепарация спектральных линий.** Рассмотрим вопрос о разделении (сепарации) перекрывающихся спектральных линий на примере, представленном на рис. 31.

В Приложении представлены головная программа `Overlap_gauss3.m`, а также  $m$ -функции `gauss.m`, `Deriv_z.m`, `Codesc.m`, `F1.m`, `deltap.m`, предназначенные для сепарации трех гауссовых спектральных линий (см. рис. 31–34).

Сначала оценим число компонент  $N$  в примере на рис. 31, для чего используем метод производных. Численно с помощью  $m$ -функции `diff.m` находим производные по  $\lambda$  1-го порядка  $Z'(\lambda)$  и 2-го порядка  $Z''(\lambda)$  суммарного спектра  $Z(\lambda)$ . На рис. 32а приведен спектр (исходная функция)  $Z(\lambda)$ , зашумленный 5-процентным шумом. Первая производная  $Z'(\lambda)$  от зашумленного спектра  $Z(\lambda)$  дает очень неустойчивый результат (пунктир на рис. 32б). Еще более неустойчивый результат даёт вторая производная  $Z''(\lambda)$  (пунктир на рис. 32в).

Чтобы сделать процедуру дифференцирования устойчивой, мы аппроксимировали зашумленный спектр  $Z(\lambda)$  сглаживающим кубическим сплайном

[2, 12] с помощью  $m$ -функции `csaps.m` (ср. [44, 45]) и затем дважды дифференцировали сплайн. Заметим, что степень сглаживания сплайна регулируется параметром сглаживания  $P \in [0, 1]$ , причем при  $P = 0$  сплайн получается максимально гладким, а при  $P = 1$  сглаживание отсутствует и сплайн становится интерполирующим.

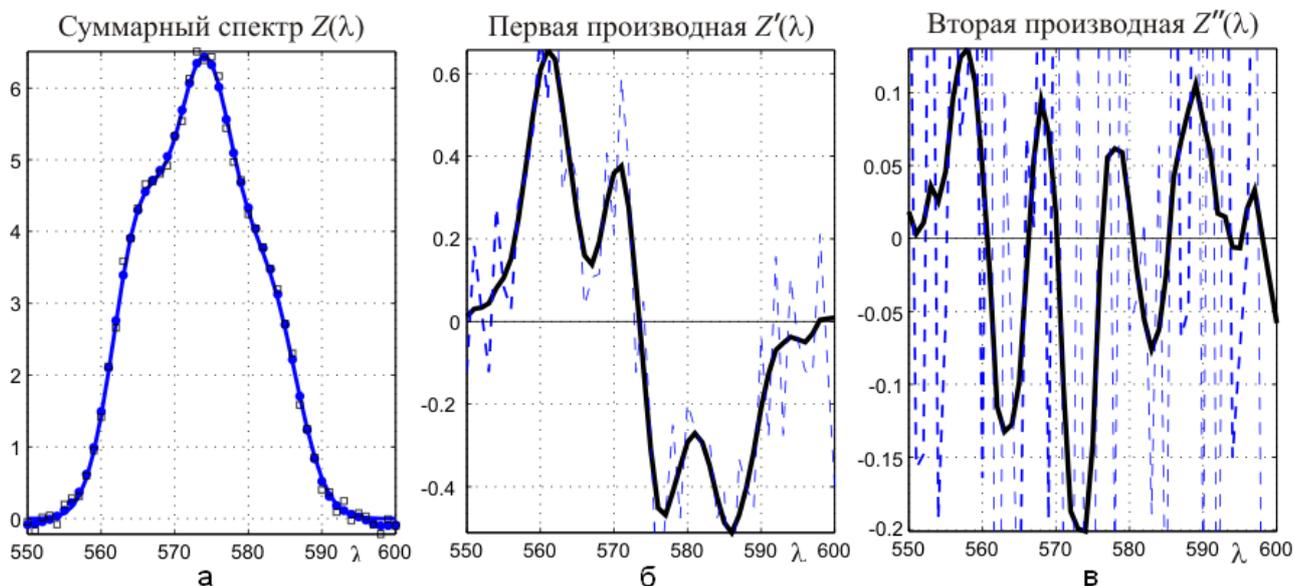


Рис. 32. Метод производных. а – суммарный спектр  $Z(\lambda)$  незашумленный (непрерывная линия), зашумленная (квадратики) и сглаживающий сплайн (точки); б – 1-я производная  $Z'(\lambda)$  от зашумленной функции  $Z(\lambda)$  (пунктирная линия) и от сплайна (непрерывная линия); в – 2-я производная  $Z''(\lambda)$  от зашумленной  $Z(\lambda)$  (пунктирная линия) и от сплайна (непрерывная линия)

Способом подбора выбран умеренный параметр сглаживания:  $P = 0.4$ . Первая производная от сплайна (рис. 32б) имеет одно нулевое значение при  $\lambda = 574$  нм, а вторая производная (рис. 32в) – три отрицательных экстремума при  $\lambda = 563, 573, 583$  нм. Сделан вывод, что число линий  $N = 3$  и их центральные длины волн  $\bar{\lambda} = 563, 573, 583$  нм (точные значения  $\bar{\lambda} = 565, 574, 583$  нм).

**Уточнение параметров линий.** После получения оценки количества компонент  $N$  на основе 1-й и 2-й производных переходим к уточнению параметров компонент. Компоненты  $z_j(\lambda)$ ,  $j = \overline{1, N}$ , суммарного спектра  $Z(\lambda)$  моделируем гауссианами согласно (60), у каждой по три искомым параметра: амплитуда  $A_j$ , координата максимума  $\bar{\lambda}_j$  и СКО  $\sigma_j$ .

Далее  $3N$  параметров объединяем в единый вектор  $p$  длиной  $3N$ :

$$p = [A_1, \bar{\lambda}_1, \sigma_1, A_2, \bar{\lambda}_2, \sigma_2, A_3, \bar{\lambda}_3, \sigma_3]. \quad (72)$$

Значения  $p_J$ ,  $J = \overline{1, 3N}$  находим путем минимизации функционала невязки

$$F = \sum_{i=1}^n (\tilde{Z}_i - Z_i)^2 \quad (73)$$

с ограничениями на параметры  $p_J$  в виде:

$$p_{\min_J} \leq p_J \leq p_{\max_J}, \quad J = \overline{1, 3N}, \quad (74)$$

где  $\tilde{Z}_i = \tilde{Z}(\lambda_i)$  – измеренные значения суммарного спектра, а  $Z_i = Z_i(p)$  – рассчитанные значения суммарного спектра.

Для минимизации функционала (73) используем *метод координатного спуска* (method of coordinate descent) [39]. В этом методе ограничения не позволяют выходить решению за пределы «коридора», даваемого неравенствами (74), тем самым, обеспечивая устойчивость и сходимость решения. В работе [39] приведен псевдокод метода координатного спуска, а в Приложении – m-функция Codesc.m для его реализации.

Выполнено уточнение параметров трёх линий-гауссиан путем минимизации функционала (73) методом координатного спуска (КС) с ограничениями типа (74):  $p_{\min} = [3.3, 560, 3.3, 5.6, 572, 4, 2.6, 580, 3.7]$ ,  $p_{\max} = [4, 568, 3.9, 6.5, 576, 4.3, 3.1, 586, 4.3]$ . Методом КС выполнено 18 итераций и получено следующее решение:  $p = [3.715, 565.0, 3.666, 6.068, 574.0, 4.163, 2.862, 582.9, 3.880]$ . При этом достигнуто следующее значение функционала:  $F = 0.457$ .

Вычислена среднеквадратическая погрешность решения  $p$  согласно [39]:

$$\delta p = \left[ \frac{1}{3N} \sum_{J=1}^{3N} w_J (p_J - \bar{p}_J)^2 \right]^{1/2}, \quad (75)$$

где  $p_J$  – вычисленные значения параметров линий,  $\bar{p}_J$  – точные параметры (известные лишь при обработке модельных спектров), а  $w_J$  – веса, равные  $w_J = 1/p_{\text{mid}_J}^2$ , причем  $p_{\text{mid}_J} = (p_{\min_J} + p_{\max_J})/2$ . Погрешность  $\delta p$  решения  $p$  методом КС по формуле (75) получилась равной 0.0103, т.е. 1%.

На рис. 33 – результат восстановления трёх линий-гауссиан методом КС.

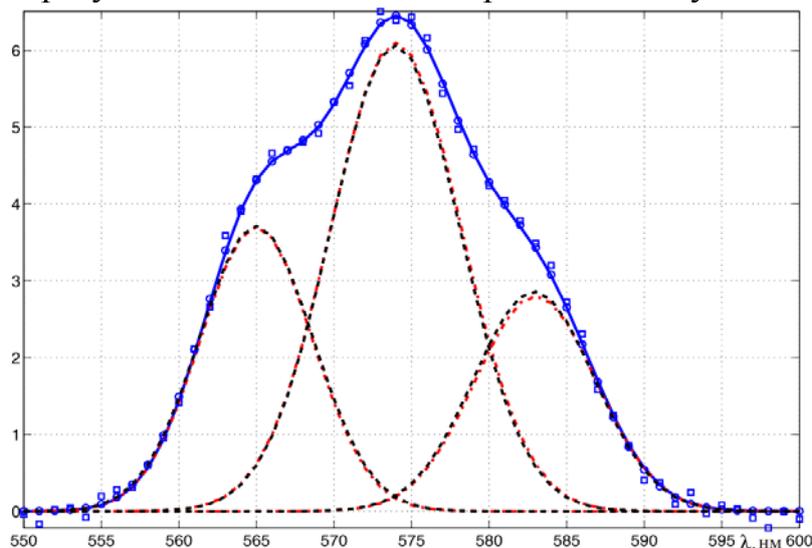


Рис. 33. Определение параметров трёх линий-гауссиан методом КС. Непрерывная линия – точный суммарный профиль  $Z(\lambda)$ , квадратики – зашумленный профиль, пунктирные линии – найденные профили трёх линий, кружки – суммарный найденный профиль, красный штрих-пунктир – точные профили линий

Для сравнения было выполнено также уточнение параметров трёх линий-гауссиан путем минимизации функционала (73) симплекс-методом Нелдера–Мида (НМ) [46], реализованным в *m*-функции `fminsearch.m` [46, с. 399]. Специфика метода НМ состоит в том, что он не использует ограничения, как метод КС, а использует начальные приближения для параметров. Кроме того, метод НМ (как и КС) не требует вычисления градиентов.

В методе НМ использовано начальное приближение для вектора  $p$  согласно (72) в виде:  $p = [3, 563, 3.3, 6, 573.5, 4.4, 3, 583, 3.8]$ . Методом НМ было выполнено 1098 итераций и получено уточнённое решение:  $p = [4.053, 565.3, 3.728, 5.981, 574.0, 3.755, 3.215, 582.5, 3.890]$ . При этом  $F = 0.372$ , а погрешность решения  $\delta p = 0.0697$  (примерно в 6–7 раз больше, чем методом КС).

На рис. 34 – результат восстановления трёх линий-гауссиан методом НМ.

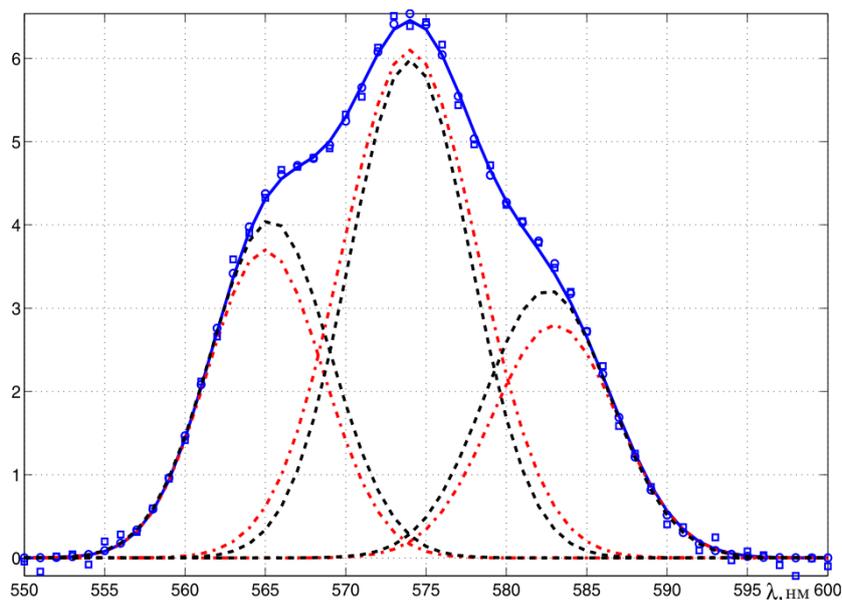


Рис. 34. Определение параметров трёх линий-гауссиан путем минимизации функционала  $F$  методом Нелдера–Мида. Обозначения такие же, как на рис. 33

Методом сепарации были обработаны также другие спектры:

– даны (без шума) три спектральных перекрывающихся линии, суммарный спектр которых сглажен аппаратной функцией, в результате сначала устраняется сглаживание, а потом разделяются линии [40];

– даны (с шумом) две линии-гауссианы, а также лоренцианы с производными от спектров вплоть до 4-го порядка [41].

Решение всех рассмотренных вариантов [39–41] показало эффективность методики разделения линий на основе дифференцирования спектра с использованием сглаживающих сплайнов и последующего уточнения параметров линий.

Изложенная методика разделения (сепарации) перекрывающихся спектральных линий может быть использована для разрешения близких линий в спектре, для восстановления тонкой структуры отдельных линий и т.д., тем самым, для повышения разрешающей способности спектрометров за счёт математической и компьютерной обработки спектров.

**Метод сверток.** Остановимся ещё на одном методе разделения линий.

В работе [43] для разделения перекрывающихся спектральных линий предложен и реализован так называемый «Метод сверток экспериментальных спектров с производными гауссовой функции». Он заключается в следующем.

Искомые (восстанавливаемые) пики-линии аппроксимируются гауссианами, что характерно для масс-спектрометрических пиков (в наших обозначениях):

$$z(\lambda) = A \exp\left[-\frac{(\lambda - \lambda_{\max})^2}{2\sigma^2}\right], \quad (76)$$

где  $A$  – амплитуда пика;  $\sigma$  – СКО, причем  $\mu = \sqrt{2\ln 2}\sigma = 1.1773\sigma$  – полуширина пика;  $\lambda_{\max}$  – положение вершины пика. Суммарный (измеренный) спектр равен суперпозиции отдельных пиков (76):

$$Z(\lambda) = \sum_{j=1}^N z_j(\lambda), \quad a \leq \lambda \leq b, \quad (77)$$

где  $j$  – номер пика,  $N$  – число пиков,  $[a, b]$  – пределы суммарного спектра.

Далее вводится *гауссовая функция* с единичной амплитудой, с СКО  $\sigma_0$  (и полушириной  $\mu_0$ ) и с нулевым смещением (положением вершины пика):

$$z_0(\lambda) = \exp\left(-\frac{\lambda^2}{2\sigma_0^2}\right) = \exp\left[-\frac{1}{2}(\lambda/\sigma_0)^2\right]. \quad (78)$$

Гауссовая функция  $z_0(\lambda)$  играет роль *базисной функции*.

Затем вычисляются аналитически производные 2-го и 4-го порядков от  $z_0(\lambda)$  по  $\lambda/\sigma_0$  [43, 38]:

$$z_0''(\lambda) = \frac{d^{(2)}z_0(\lambda)}{d(\lambda/\sigma_0)^{(2)}} = -\left[1 - (\lambda/\sigma_0)^2\right] \cdot \exp\left[-\frac{1}{2}(\lambda/\sigma_0)^2\right], \quad (79)$$

$$z_0^{IV}(\lambda) = \frac{d^{(4)}z_0(\lambda)}{d(\lambda/\sigma_0)^{(4)}} = \left[3 - 6(\lambda/\sigma_0)^2 + (\lambda/\sigma_0)^4\right] \cdot \exp\left[-\frac{1}{2}(\lambda/\sigma_0)^2\right]. \quad (80)$$

Далее выполняются свертки (convolution) производных  $z_0''(\lambda)$  и  $z_0^{IV}(\lambda)$  с одним из пиков (76):

$$c_2(\lambda) = -A \int_{-\infty}^{\infty} z_0''(\lambda - l) \exp\left(-\frac{l^2}{2\sigma^2}\right) dl, \quad (81)$$

$$c_4(\lambda) = -A \int_{-\infty}^{\infty} z_0^{IV}(\lambda - l) \exp\left(-\frac{l^2}{2\sigma^2}\right) dl. \quad (82)$$

Вычислив аналитически интегралы, получим:

$$c_2(\lambda) = \frac{A\sigma\sqrt{\pi}}{[1+(\sigma/\sigma_0)^2]^{3/2}} \left(1 - \frac{\lambda^2}{\sigma^2 + \sigma_0^2}\right) \exp\left(-\frac{\lambda^2}{\sigma^2 + \sigma_0^2}\right), \quad (83)$$

$$c_4(\lambda) = \frac{A\sigma\sqrt{\pi}}{[1+(\sigma/\sigma_0)^2]^{5/2}} \left(3 - \frac{6\lambda^2}{\sigma^2 + \sigma_0^2}\right) + \frac{\lambda^4}{(\sigma^2 + \sigma_0^2)^2} \exp\left(-\frac{\lambda^2}{\sigma^2 + \sigma_0^2}\right). \quad (84)$$

Из формул (83) и (84) следует, что  $c_2(\lambda)$  и  $c_4(\lambda)$  при  $\lambda=0$  имеют максимальные значения:

$$c_2(0) = c_{2\max} = \frac{A\sigma\sqrt{\pi}}{[1+(\sigma/\sigma_0)^2]^{3/2}}, \quad (85)$$

$$c_4(0) = c_{4\max} = \frac{3A\sigma\sqrt{\pi}}{[1+(\sigma/\sigma_0)^2]^{5/2}}. \quad (86)$$

Используя значения  $c_2(0) \equiv C_2$  и  $c_4(0) \equiv C_4$ , можно найти параметры  $\sigma$  и  $A$  пика (76):

$$\sigma = \sigma_0 \left(3 \frac{C_2}{C_4} - 1\right)^{1/2}, \quad (87)$$

$$A = \frac{C_2 [1+(\sigma/\sigma_0)^2]^{3/2}}{\sqrt{\pi} \sigma}. \quad (88)$$

Практическую реализацию рассмотрим на примере, заимствованном из [43] (см. рис. 35).

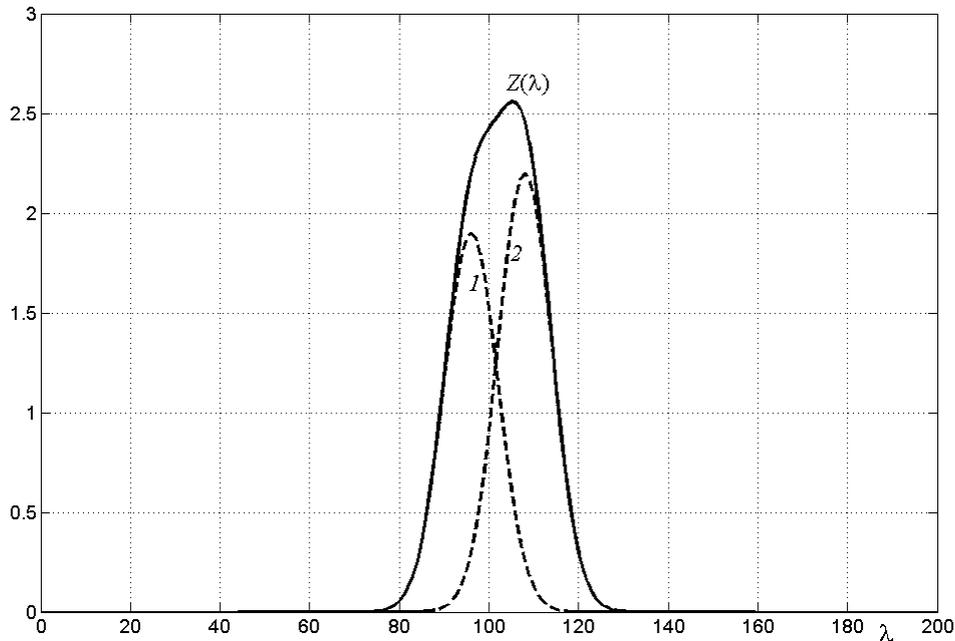


Рис. 35. Суммарный спектр  $Z(\lambda)$ , состоящий из двух пиков. 1 –  $z_1(\lambda)$ ,  $A=1.9$ ,  $\lambda_{\max}=96$ ,  $\sigma=6$ ; 2 –  $z_2(\lambda)$ ,  $A=2.2$ ,  $\lambda_{\max}=108$ ,  $\sigma=6$

На практике отдельные пики  $z(\lambda)$  не измеряются и свертки (81) и (82) не реализуемы, а измеряется суммарный спектр  $Z(\lambda)$ , поэтому делаем свертки  $z_0''(\lambda)$  и  $z_0^{IV}(\lambda)$  с  $Z(\lambda)$ :

$$c_2(\lambda) = - \int_{-\infty}^{\infty} z_0''(\lambda - l) Z(l) dl, \quad (89)$$

$$c_4(\lambda) = - \int_{-\infty}^{\infty} z_0^{IV}(\lambda - l) Z(l) dl. \quad (90)$$

На рис. 36 и 37 – результаты сверток (89) и (90):

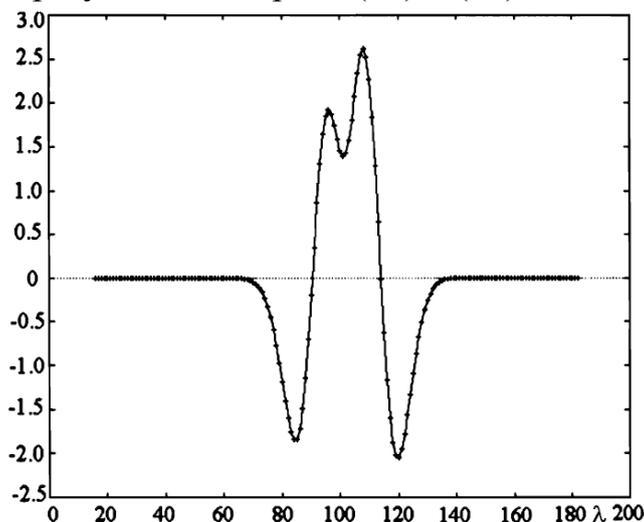


Рис. 36. Свертка второй производной  $z_0''(\lambda)$  с  $Z(\lambda)$

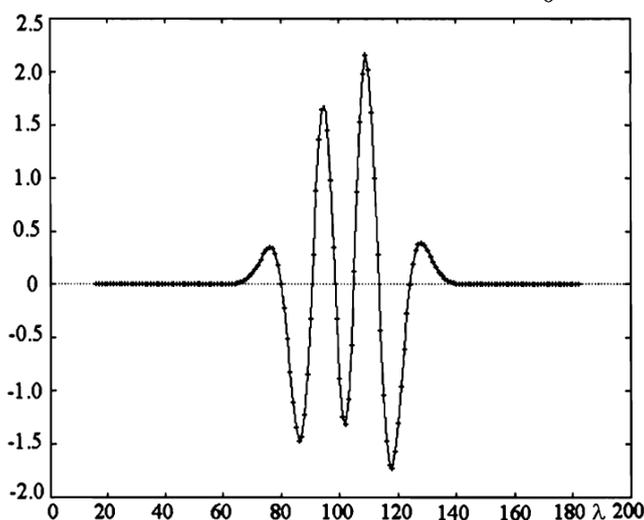


Рис. 37. Свертка четвертой производной  $z_0^{IV}(\lambda)$  с  $Z(\lambda)$

Видим, что свертки 2-й и 4-й производных дают по два положительных экстремума при практически точных значениях  $\lambda_{\max}$  (высокая точность обусловлена неучетом шумов).

Далее определялись параметры  $\sigma$  и  $A$  обоих пиков с использованием формул (87) и (88).

В заключение эти параметры уточнялись с помощью итерационной процедуры, использующей последовательные вычеты найденных пиков из суммарного спектра  $Z(\lambda)$  (см. [43]).

На рис. 38 – окончательный результат разделения пиков.

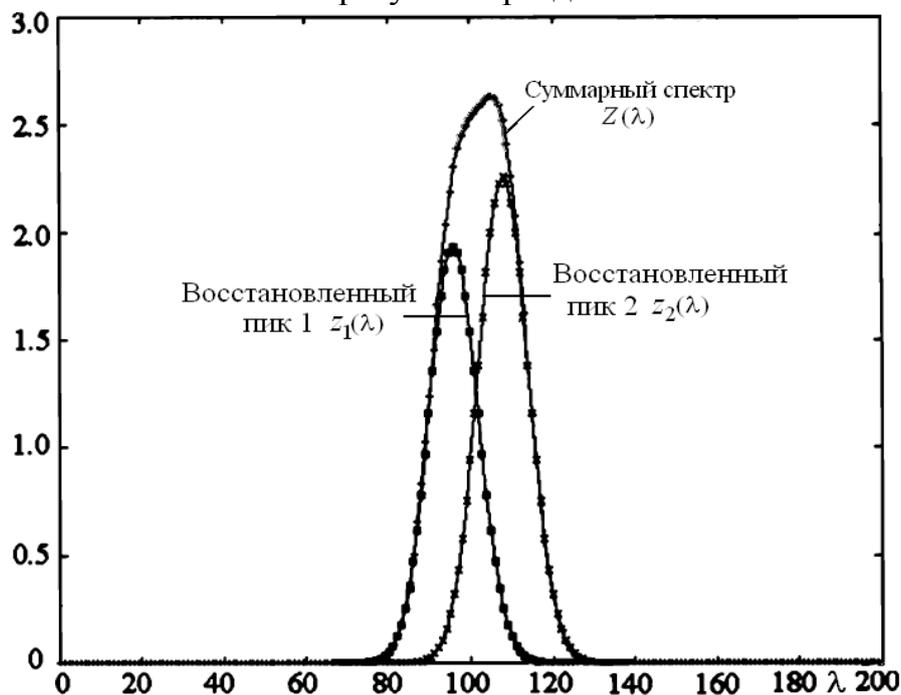


Рис. 38. Восстановленные пики

Отметим следующие достоинства методики, изложенной в [43, 38]:

1. Не нужно численно дифференцировать экспериментальный спектр  $Z(\lambda)$ , тем более, зашумленный  $\tilde{Z}(\lambda)$ , что связано обычно с большими погрешностями (если не применять специальные процедуры, например, сглаживающие сплайны [39, 41]). Вместо этого нужно брать аналитически производные (2-го и 4-го порядков) от базисной функции  $z_0(\lambda)$  [43, 38], что выполняется точно.

2. Экспериментальный спектр  $Z(\lambda)$  через посредство свертки входит под интегралы (89) и (90), а интегрирование (в противоположность дифференцированию) обладает сглаживающими свойствами, другими словами, фильтрует шумы, что хорошо продемонстрировано в статье [47, рис. 9].

3. В методике предложена итерационная процедура для уточнения параметров  $A$  и  $\sigma$  пиков путем последующих вычетов найденных пиков из суммарного спектра  $Z(\lambda)$  [43].

Отметим также недостатки методики:

1. Не использованы производные 1-го и 3-го порядков, которые также указывают на пики (при  $c_1(\lambda) = c_3(\lambda) = 0$ ).

2. В методике не учтено, что при числе пиков  $N \geq 2$  свертка получается не от отдельных пиков, а от суммарного спектра  $Z(\lambda)$ , что ведет к погрешности параметра  $\lambda_{\max}$ . При этом параметр  $\lambda_{\max}$  не уточняется в итерационной процедуре и требуется специальный способ для уточнения  $\lambda_{\max}$ .

## 5. Приложение (некоторые MatLab-программы)

В данном приложении приведены тексты (листинги) MatLab-программ: головной программы `Overlap_gauss3.m`, а также `m`-функций `gauss.m`, `Deriv_z.m`, `Codesc.m`, `F1.m`, `deltar.m`, предназначенных для сепарации трех гауссовых спектральных линий (см. рис. 31–34).

```
%% Overlap_gauss3.m, Сизиков, Лавров, головная программа, MatLab.
%Обратная задача спектроскопии:
%сепарация (разделение) перекрывающихся (overlapped) гауссовых линий.
%Сначала выполняется решение прямой задачи - расчет суммарного спектра
%Ze(lambda) в виде суммы 3 линий-гауссиан z1, z2 и z3 (gauss.m) и
%зашумление спектра Zel(lambda) с помощью датчика случайных чисел randn.
%Затем - восстановление (сепарация) отдельных линий z1, z2 и z3
%методом координатного спуска с использованием способа сужающихся
%ограничений (функция CoDesc.m), а также путем минимизации функционала
%F1 симплекс-методом Нелдера-Мида (без ограничений, но с начальным
%приближением) (функция fminsearch).
clc, clear, clear global
%% Параметры задачи:
a=550; b=600; %пределы по lambda в нм
h=1; %h - шаг дискретизации по lambda в нм
n=(b-a)/h+1; %n=51 - число узлов по lambda
lambda = a:h:b; %сетка по lambda 1xn double с шагом h
deltaz = 0.1; %погрешность измеренного суммарного спектра (5%)
%% Прямая задача
%Точные (exact) параметры отдельных линий в виде единого вектора pe:
pe = [3.7,565,3.6, 6.1,574,4.2, 2.8,583,3.9]; %точные параметры
%Степени наложения пиков:
tau12=1.1773*(pe(3)+pe(6))/2; tau23=1.1773*(pe(6)+pe(9))/2; %полуширины
Delta12=pe(5)-pe(2); Delta23=pe(8)-pe(5); %расстояния между пиками
q12=1+abs(pe(1)-pe(4))/(pe(1)+pe(4));
q23=1+abs(pe(4)-pe(7))/(pe(4)+pe(7));
D12=tau12/Delta12*q12; D23=tau23/Delta23*q23; %степени наложения пиков
D=(D12+D23)/2; %0.68 - среднее значение степени наложения пиков
%Профили отдельных линий:
z1 = gauss(pe(1), lambda, pe(2), pe(3)); %z1 1xn double
z2 = gauss(pe(4), lambda, pe(5), pe(6)); %z2 1xn double
z3 = gauss(pe(7), lambda, pe(8), pe(9)); %z3 1xn double
%Суммарный спектр Z(lambda):
Ze = z1+z2+z3; %Ze 1xn double
%% Зашумленный экспериментальный суммарный спектр zel(lambda):
randn('state',0) %инициализация генератора случайных чисел
Zel = Ze+deltaz*randn(1,n); %добавление нормальной погрешности
%с СКО=deltaz=0.1 (5%) и MO=0
%% 2D-графика (исходные линии, их сумма и измеренный спектр):
figure
plot(lambda,z1,'-r', lambda,z2,'-r', lambda,z3,'-r',...
      lambda,Ze,'-b', 'LineWidth',2.5)
grid on %сетка
hold on %продолжение графика с новыми параметрами
plot(lambda,Zel,'sb','LineWidth',1.5,'MarkerSize',6)
hold on, text(596.15,-.4,'\lambda','FontSize',18)
text(597.2,-.42,' нм','FontName','Times New Roman','FontSize',16)
text(566.25,3.65,'\it1','FontSize',16,'color','k')
text(571.3,4.5,'\it2','FontSize',16,'color','k')
```

```

text(582.5,2.55,'\it3','FontSize',16,'color','k')
text(577.8,5.5,'\it4','FontSize',16,'color','k')
Zmin = min(Ze1); Zmax = max(Ze1); xlim([a b]); ylim([Zmin Zmax+1e-6]);
hold off %окончание построения графика
%Сглаживание Ze1 кубическим сглаживающим сплайном:
Zels = csaps(lambda,Ze1,0.4,lambda); %0.4 - параметр сглаживания
%Производные 1-го (D1) и 2-го (D2) порядков от Zels(lambda):
[D1,D2] = Deriv_z(Ze,Ze1,Zels,lambda); %с 2D-графикой внутри Deriv_z
%% Поиск 9 параметров трех линий путем минимизации функционала
% методом координатного спуска с ограничениями:
pmin = [3.3,560,3.3, 5.6,572,4.0, 2.6,580,3.7];
pmax = [4.0,568,3.9, 6.5,576,4.3, 3.1,586,4.3];
e = (pmax-pmin)/100;
[pc,Fmin,Niter] = CoDesc(@F1,pmin,pmax,e,100,Ze1,lambda)
%Профили трех рассчитанных линий:
zc1 = gauss(pc(1), lambda, pc(2), pc(3)); %1xn double
zc2 = gauss(pc(4), lambda, pc(5), pc(6)); %1xn double
zc3 = gauss(pc(7), lambda, pc(8), pc(9)); %1xn double
Zc = zc1+zc2+zc3; %суммарный профиль 1xn double
%% 2D-графика:
figure
plot(lambda,Ze,'-b',lambda,z1,'-.r',lambda,z2,'-.r',...
      lambda,z3,'-.r', 'LineWidth',2.5)
hold on
plot(lambda,Ze1,'sb','LineWidth',1.5,'MarkerSize',6)
hold on
plot(lambda,Zc,'ob','LineWidth',1.5,'MarkerSize',6)
hold on
plot(lambda,zc1,'--k',lambda,zc2,'--k',lambda,zc3,'--k',...
      'LineWidth',2.5)
hold on
grid on %сетка
text(596.15,-.4,'\lambda','FontSize',18)
text(597.2,-.42,' нм','FontName','Times New Roman','FontSize',16)
Zmin = min(Ze1); Zmax = max(max(Ze1),max(Zc));
xlim([a b]), ylim([Zmin Zmax+1e-6])
hold off
%% Поиск 9 параметров трех линий путем минимизации функционала
%симплекс-методом Нелдера-Мида без ограничений, но с заданием
%начального приближения для параметров:
options = optimset('tolX',1e-4); %задание погрешности переменных
[p,opt,rosexflag,rosout] = fminsearch(@F1(p,Ze1,lambda),...
[3,563,3.3,6,573.5,4.4,3,583,3.8], options) %нач. приближение вектора p
%p - искомый вектор длиной 9
%(амплитуда, координата и sigma каждой линии).
%Профили трех рассчитанных линий:
zm1 = gauss(p(1), lambda, p(2), p(3)); %1xn double
zm2 = gauss(p(4), lambda, p(5), p(6)); %1xn double
zm3 = gauss(p(7), lambda, p(8), p(9)); %1xn double
Zm = zm1+zm2+zm3; %суммарный профиль 1xn double
%% 2D-графика:
figure
plot(lambda,Ze,'-b',lambda,z1,'-.r',lambda,z2,'-.r',...
      lambda,z3,'-.r', 'LineWidth',2.5)
hold on
plot(lambda,Ze1,'sb','LineWidth',1.5,'MarkerSize',6)

```

```

hold on
plot(lambda,Zm,'ob','LineWidth',1.5,'MarkerSize',6)
hold on
plot(lambda,zm1,'--k',lambda,zm2,'--k',lambda,zm3,'--k',...
      'LineWidth',2.5)
hold on
grid on %сетка
text(596.15,-.4,'\lambda','FontSize',18)
text(597.2,-.42,' нм','FontName','Times New Roman','FontSize',16)
Zmin = min(Ze1); Zmax = max(max(Ze1),max(Zm));
xlim([a b]), ylim([Zmin Zmax+1e-6])
hold off;
%% Среднеквадратическая погрешность решения pc методом
%координатного спуска относительно точного решения pe:
pmid = (pmin+pmax)/2; w = 1./(pmid.^2); %w - веса
dpc = deltap(pc,pe,w); %=0.0103
%Среднеквадратическая погрешность решения p методом
%Нелдера-Мида относительно точного решения pe:
dp = deltap(p,pe,w); %=0.0697

%% gauss.m, Сизиков, Лавров, MatLab.
%Спектральная линия в виде гауссианы.
function g = gauss(A,lambda,lambda0,sigma) %собственная m-функция
%Входные параметры (аргументы):
%A - амплитуда (скаляр), lambda - текущая длина волны (массив),
%lambda0 - центральная длина волны (скаляр),
%sigma - СКО (скаляр) (sigma^2 - дисперсия).
%Выходной параметр (аргумент):
%gauss - гауссиана (g - посредник) - массив длины lambda.
g = A*exp(-(lambda-lambda0).^2/(2*sigma^2));

%% Deriv_z.m, Сизиков, Лавров, m-функция, MatLab.
%Вычисление производных 1-го (D1) и 2-го (D2) порядков от z(lam).
function [D1,D2] = Deriv_z(ze,ze1,zs,lam) %собственная m-функция
%Входные параметры (аргументы):
%ze - суммарный незашумленный спектр,
%ze1 - суммарный зашумленный спектр,
%zs - суммарный зашумленный спектр, сглаженный сплайном,
%lam - массив значений lambda.
%Выходные параметры (аргументы):
%D1 и D2 - производные 1-го и 2-го порядков от ze1(lam)
%(вычисляются также производные D1s и D2s от zs(lam)).
%Параметры:
n=length(lam); n1=n-1; n2=n-2; n3=n-3;
a=lam(1); b=lam(n); h=(b-a)/n1;
%Исходная функция:
figure, subplot(1,3,1)
plot(lam,ze,'-b',lam,zs,'.b','LineWidth',2.0,'MarkerSize',14)
hold on
plot(lam,ze1,'sk','LineWidth',1.0,'MarkerSize',4)
title('Суммарный спектр Z(\lambda)','FontSize',12)
text(594,-0.5,'\lambda','FontSize',14)
zmin = min(ze1); zmax = max(ze1); xlim([a b]), ylim([zmin zmax+1e-6])
grid on %сетка
hold off
%Первая производная:
D1=diff(ze1)/h; D1(n)=2*D1(n1)-D1(n2); %экстраполяция

```

```

D1s=diff(zs)/h; D1s(n)=2*D1s(n1)-D1s(n2); %экстраполяция
subplot(1,3,2), plot(lam,D1s,'-k','LineWidth',2.5), hold on;
plot(lam,D1,'--b',lam,zeros(1,n),'-k','LineWidth',1.25)
title('Первая производная Z\prime(\lambda)','FontSize',12)
D1smin = min(D1s); D1smax = max(D1s);
text(594,D1smin-0.06,'\lambda','FontSize',14)
xlim([a b]), ylim([D1smin D1smax]), grid on, hold off
%Вторая производная:
D2=diff(D1)/h; D2(n1)=2*D2(n2)-D2(n3); D2(n)=2*D2(n1)-D2(n2);
D2s=diff(D1s)/h; D2s(n1)=2*D2s(n2)-D2s(n3); D2s(n)=2*D2s(n1)-D2s(n2);
subplot(1,3,3), plot(lam,D2s,'-k','LineWidth',2.5), hold on
plot(lam,D2,'--b',lam,zeros(1,n),'-k','LineWidth',1.25)
title('Вторая производная Z\prime\prime(\lambda)','FontSize',12)
D2smin = min(D2s); D2smax = max(D2s);
text(594,D2smin-0.015,'\lambda','FontSize',14)
xlim([a b]), ylim([D2smin D2smax]), grid on, hold off

%% CoDesc.m, Сизиков, Лавров, m-функция, MatLab
%Определение N>=1 переменных (параметров)
%путём минимизации функционала (функции) F(p1,p2,...,pN)
%методом координатного спуска (method of Coordinate Descent)
%с ограничениями pmin(j) <= p(j) <= pmax(j), j=1,2,...,N.
function [p,Fmin,Niter] = CoDesc(F,pmin,pmax,e,s,z,x)
%Входные параметры (аргументы):
%F - функционал,
%рmin и рmax - ограничения на параметры (массивы длиной N),
%e - задаваемые погрешности определения p (массив длиной N),
% например, e(j) = (рmax(j)-рmin(j))/100,
%s>=3 задаёт начальные шаги h(j)=(рmax(j)-рmin(j))/s,
%z - экспериментальный суммарный массив, например,
% спектр z(x) длиной n,
%x - сетка длиной n.
%Выходные параметры (аргументы):
%р - найденные искомые параметры (массив длиной N),
%Fmin - полученный минимум функционала (число),
%Niter - общее число выполненных итераций.
N = length(pmin); %длины векторов p,рmin,рmax,e
h = (рmax-рmin)/s; d=h; %начальные шаги
p = (рmin+рmax)/2; %начальные p(j)
%% Итерации:
repeat = true; Niter = 0;
while(repeat)
    for j=1:N %j - номер параметра
        Niter = Niter+1; fj = F(p,z,x);
        while (abs(h(j))>=e(j)/3)
            gj=fj; p(j)=p(j)+h(j);
            if(p(j)<рmin(j)) %в случае выхода за левое ограничение
                p(j)=рmin(j); h(j)=-h(j)/3; %уменьшение шага и смена его знака
                continue;
            end %if
            if(p(j)>рmax(j)) %в случае выхода за правое ограничение
                p(j)=рmax(j); h(j)=-h(j)/3; %уменьшение шага и смена его знака
                continue;
            end %if
            fj = F(p,z,x);
            if (fj>=gj)
                h(j)=-h(j)/3;

```

```

        end %if
    end %while
    h(j)=sign(h(j))*d(j); %восстановление шага со знаком
end %j
h = h/9; d=h; %уменьшение всех шагов
repeat = false;
for j=1:N
    if(abs(d(j))>=e(j)/9)
        repeat = true;
        break;
    end %if
end %j
end %while
Fmin = fj;

%% F1.m, Сизиков, Лавров, m-функция, MatLab.
%1) Моделируется спектр z в виде суммы трёх
% спектральных линий-гауссиан или лоренциан.
%2) Вычисляется функционал-невязка относительно N=9 параметров:
% p1=A1, p2=lambda0_1, p3=sigma1, p4=A2, ...
function f = F1(p, ze1, lambda)
%Входные параметры (аргументы):
%p - начальное приближение вектора p 1xN double,
%ze1 - суммарный экспериментальный спектр ze1(lambda) длиной n,
%lambda - сетка по lambda с шагом h длиной n.
%Выходной параметр:
%F1 - функционал (f - посредник).
%% Линии-гауссианы (такой же длины, как lambda):
z1 = gauss(p(1), lambda, p(2), p(3)); %1xn double
z2 = gauss(p(4), lambda, p(5), p(6)); %1xn double
z3 = gauss(p(7), lambda, p(8), p(9)); %1xn double
z = z1+z2+z3; %суммарный рассчитанный спектр 1xn double
f = sum((ze1-z).^2); %функционал (скаляр)

%% deltap.m, Сизиков, Лавров, m-функция, MatLab.
%Вычисление среднеквадратической погрешности решения p
%относительно точного решения pe.
function d = deltap(p, pe, w) %собственная m-функция
%Входные параметры (аргументы):
%p - некоторое решение (double),
%pe - точное решение (double),
%w - веса (double)
%(p, pe и w - одинаковой длины 3n).
%Выходной параметр (аргумент):
%deltap - среднекв. погрешность (double) (d - посредник).
n3 = length(p);
d = sqrt(sum(w.*((p-pe).^2))/n3);

```

## Список использованной литературы

1. Сизиков В.С. Прямые и обратные задачи восстановления изображений, спектроскопии и томографии с MatLab. Учеб. пособие. – СПб.: Лань, 2017. 412 с.
2. Сизиков В.С. Математические методы обработки результатов измерений. Учебник для вузов. – СПб.: Политехника, 2001. 240 с.
3. Сизиков В.С. Обратные прикладные задачи и MatLab. Учеб. пособие. – СПб.: Лань, 2011. 256 с.
4. Бейтс Р., Мак-Доннелл М. Восстановление и реконструкция изображений. – М.: Мир, 1989. 336 с.
5. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2006. 1072 с.
6. Дьяконов В., Абраменкова И. MATLAB. Обработка сигналов и изображений. Специальный справочник. – СПб.: Питер, 2002. 608 с.
7. Яне В. Цифровая обработка изображений. – М.: Техносфера, 2007. 584 с.
8. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображений в среде MATLAB. – М.: Техносфера, 2006. 616 с.
9. Арефьева М.В., Сысоев А.Ф. Быстрые регуляризирующие алгоритмы цифрового восстановления изображений // Вычислит. методы и программирование. 1983. Вып. 39. С. 40–55.
10. Василенко Г.И. Теория восстановления сигналов: О редукации к идеальному прибору в физике и технике. – М.: Сов. радио, 1979. 272 с.
11. Тихонов А.Н., Арсенин В.Я. Методы решения некорректных задач. 3-е изд. – М.: Наука, 1986. 288 с.
12. Воскобойников Ю.Е., Преображенский Н.Г., Седельников А.И. Математическая обработка эксперимента в молекулярной газодинамике. – Новосибирск: Наука, 1984. 240 с.
13. Тихонов А.Н., Гончарский А.В., Степанов В.В. Обратные задачи обработки фотоизображений // Некорректные задачи естествознания / Под ред. А.Н. Тихонова, А.В. Гончарского. – М.: Изд-во МГУ, 1987. С. 185–195.
14. Petrov Yu.P., Sizikov V.S. Well-Posed, Ill-Posed, and Intermediate Problems with Applications. – Leiden–Boston: VSP, 2005. 234 pp.
15. Верлань А.Ф., Сизиков В.С. Интегральные уравнения: методы, алгоритмы, программы. – Киев: Наук. думка, 1986. 544 с.
16. Russ J.C. The Image Processing Handbook. 5th ed. – Boca Raton: CRC Press, 2007. 818 pp.
17. Бакушинский А.Б., Гончарский А.В. Некорректные задачи. Численные методы и приложения. – М.: Изд-во МГУ, 1989. 199 с.
18. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах. Учеб. пособие. – Новосибирск: Изд-во НГТУ, 2002. 352 с.
19. Сизиков В.С. Спектральный способ оценки функции рассеяния точки в задаче устранения искажений изображений // Оптический журнал. 2017. Т. 84.

№ 2. С. 36–44.

20. Сизиков В.С., Степанов А.В., Меженин А.В., Бурлов Д.И., Экземпляров Р.А. Определение параметров искажений изображений спектральным способом в задаче обработки снимков поверхности Земли, полученных со спутников и самолётов // *Оптический журнал*. 2018. Т. 85. № 4. С. 19–27.
21. Lim J.S. *Two-dimensional Signal and Image Processing*. – New Jersey: Prentice Hall PTR, 1990. 694 pp.
22. Фисенко В.Т., Фисенко Т.Ю. Компьютерная обработка и распознавание изображений. Учеб. пособие. – СПб.: Изд-во СПбГУ ИТМО, 2008. 192 с.
23. Воскобойников Ю.Е., Мухина И.Н. Локальный регуляризирующий алгоритм восстановления контрастных сигналов и изображений // *Автометрия*. 2000, № 3. С. 45–53.
24. Сизиков В.С., Экземпляров Р.А. Последовательность операций при фильтрации шумов на искаженных изображениях // *Оптический журнал*. 2013. Т. 80, № 1. С. 39–48.
25. Воскобойников Ю.Е. Комбинированный нелинейный алгоритм восстановления контрастных изображений при неточно заданной аппаратной функции // *Автометрия*. 2007. Т. 43, № 6. С. 3–16.
26. Раутиан С.Г. Реальные спектральные приборы // *Успехи физических наук*. 1958. Т. 66. Вып. 3. С. 475–517.
27. Старков В.Н. Конструктивные методы вычислительной физики в задачах интерпретации. – Киев: Наук. думка, 2002. 264 с.
28. Физический энциклопедический словарь / Гл. ред. А.М. Прохоров. – М.: Сов. Энциклопедия, 1984. 944 с.
29. Tourin R.H., Krakow B. Applicability of infrared emission and absorption spectra to determination of hot gas temperature profiles // *Appl. Optics*. 1965. Vol. 4. No. 2. P. 237–242.
30. Ландсберг Г.С. Оптика. Учеб. пособие для вузов. 6-е изд. – М.: Физматлит, 2006. 848 с.
31. Кочиков И.В., Курамшина Г.М., Пентин Ю.А., Ягола А.Г. Обратные задачи колебательной спектроскопии. – М.: Изд-во МГУ, 1993. 204 с.
32. Глазов М.В., Болохова Т.А. Решение редуцированной проблемы Рэлея с использованием различных модификаций метода регуляризации // *Оптика и спектроскопия*. 1989. Т. 67. Вып. 3. С. 533–537.
33. Сизиков В.С., Кривых А.В. Восстановление непрерывных спектров методом регуляризации с использованием модельных спектров // *Оптика и спектроскопия*. 2014. Т. 117, № 6. С. 1040–1048.
34. Sizikov V.S., Stepanov A.V. Method of training examples in solving inverse ill-posed problems of spectroscopy // *Sci. Techn. J. ITMO (Saint-Petersburg)*. 2015. Vol. 15. No. 6. P. 1147–1154. [arxiv.org/ftp/arxiv/papers/1509/1509.05975.pdf](http://arxiv.org/ftp/arxiv/papers/1509/1509.05975.pdf)
35. Giese A.T., French C.S. The analysis of overlapping spectral absorption bands by derivative spectrophotometry // *Applied Spectroscopy*. 1955. V. 9. N 2. P. 78–96. doi: 10.1366/000370255774634089
36. Kauppinen J.K., Moffatt D.J., Mantsch H.H., Cameron D.G. Fourier self-

- deconvolution: a method for resolving intrinsically overlapped bands // *Applied Spectroscopy*. 1981. V. 35. N 3. P. 271–276. doi: 10.1366/0003702814732634
37. Михайленко В.И., Михальчук В.В. Методы разложения спектров с неразрешенной структурой // *Журнал прикладной спектроскопии*. 1987. Т. 46. № 4. С. 535–543.
  38. Манойлов В.В., Заруцкий И.В. Возможности алгоритма сверток с производными для оценки параметров масс-спектров, содержащих наложившиеся пики // *Научное приборостроение*. 2009. Т. 19. № 4. С. 103–108.
  39. Сизиков В.С., Лавров А.В. Исследование погрешностей некоторых методов разделения перекрывающихся спектральных линий в условиях воздействия помех // *Научно-технический вестник информационных технологий, механики и оптики*. 2017. Т. 17. № 5. С. 879–889. doi: 10.17586/2226-1494-2017-17-5-879-889
  40. Сизиков В.С., Лавров А.В. Сепарация непрерывных линий, взаимно перекрывающихся и сглаженных аппаратной функцией // *Оптика и спектроскопия*. 2017. Т. 123. № 5. С. 678–688. doi 10.7868/S0030403417110216
  41. Сизиков В.С., Лавров А.В. Сравнение различных методов разделения непрерывных перекрывающихся спектральных линий // *Оптика и спектроскопия*. 2018. Т. 124. № 6. С. 723–731. doi 10.21883/OS.2018.06.46071.28-18
  42. Sizikov V., Sidorov D. Discrete spectrum reconstruction using integral approximation algorithm // *Applied Spectroscopy*. 2017. V. 71. N 7. P. 1640–1651. doi: 10.1177/0003702817694181
  43. Сирвидас С.И., Заруцкий И.В., Ларионов А.М., Манойлов В.В. Обнаружение, разделение и оценка параметров масс-спектрометрических пиков методом свертки экспериментальных данных с производными гауссовых функций // *Научное приборостроение*. 1999. Т. 9. № 2. С. 71–75.
  44. Сизиков В.С. Инфракрасная томография горячего газа: математическая модель активно-пассивной диагностики // *Научно-технический вестник информационных технологий, механики и оптики*. 2013. № 6(88). С. 1–17.
  45. Sizikov V.S., Evseev V., Fateev A., Clausen S. Direct and inverse problems of infrared tomography // *Applied Optics*. 2016. V. 55. N 1. P. 208–220. doi: 10.1364/AO.55.000208
  46. Дьяконов В. MATLAB 6: Учебный курс. СПб.: Питер, 2001. 592 с.
  47. Манойлов В.В., Заруцкий И.В. Отбраковка «выбросов» и оценка параметров масс-спектрометрических сигналов для прецизионного изотопного анализа // *Научное приборостроение*. 2002. Т. 19. № 3. С. 38–46.

**Миссия университета** – генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки и технологий и других областей для содействия решению актуальных задач.

---

### **ОБРАЗОВАТЕЛЬНЫЙ ЦЕНТР ГРАФИЧЕСКИХ ТЕХНОЛОГИЙ**

Образовательный центр графических технологий возник на базе кафедры графических технологий (ГТ) – старейшей среди кафедр университета. Кафедра ГТ существует с момента образования Ленинградского института точной механики и оптики (ЛИТМО).

В период с 1930 г. кафедра претерпела множество качественных эволюционных изменений и доработок, выстроив четкий вектор текущего и дальнейшего развития, ориентированный, в первую очередь, на потребности рынка труда в аспекте компьютерной графики и информационно-коммуникационных технологий. Одним из научно-исследовательских направлений деятельности кафедры являются исследования в области восстановления искаженных изображений и спектров математическим путем с использованием компьютеров.

Перспективы развития Образовательного центра ГТ связаны с разработкой продуктов компьютерной графики и информационно-коммуникационных технологий и представлением этих продуктов на рынках компьютерной продукции.

**Сизиков Валерий Сергеевич  
Лавров Алексей Валерьевич**

**УСТОЙЧИВЫЕ МЕТОДЫ  
МАТЕМАТИКО-КОМПЬЮТЕРНОЙ  
ОБРАБОТКИ ИЗОБРАЖЕНИЙ И СПЕКТРОВ**

**Учебное пособие**

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе

**Редакционно-издательский отдел**  
**Университета ИТМО**  
197101, Санкт-Петербург, Кронверкский пр., 49