

В. С. СИЗИКОВ, К. А. КИРЬЯНОВ, Р. А. ЭКЗЕМПЛЯРОВ
**ДВА БЫСТРЫХ АЛГОРИТМА ВОССТАНОВЛЕНИЯ
 СМАЗАННЫХ ИЗОБРАЖЕНИЙ**

Представлены два быстрых алгоритма восстановления смазанного изображения, позволяющих численно решать некорректное интегральное уравнение Фредгольма I рода методом регуляризации Тихонова с использованием заранее рассчитанной („заготовленной“) матрицы. Алгоритмы сводятся к умножению этой матрицы на каждую вектор-строку смазанного изображения. Выполнено сравнение с другими быстрыми алгоритмами.

Ключевые слова: смазывание изображения, восстановление изображения, интегральное уравнение, быстрые алгоритмы, „заготовленная“ матрица.

Введение. Задача восстановления искаженных, в частности смазанных, изображений является актуальной [1—5]. Смазывание изображения объекта может произойти из-за сдвига цифрового фотоаппарата за время экспозиции, перемещения самого объекта (самолета, автомобиля), несовпадения угловой скорости вращения телескопа и небесной сферы и т.д. Восстановление смазанного изображения сводится к решению множества одномерных интегральных уравнений (ИУ) Фредгольма I рода. Задача решения таких ИУ является некорректной (существенно неустойчивой) [6], обычно используются метод регуляризации Тихонова [1, 2, 4—6], метод фильтрации Винера [3—6] и другие устойчивые методы [4].

Метод регуляризации Тихонова обычно реализуется с использованием преобразования Фурье (ПФ) в случае ИУ типа свертки, когда функция рассеяния точки (ФРТ) является пространственно-инвариантной [1—6], или с использованием квадратур в случае ИУ общего типа (когда ФРТ является пространственно-неинвариантной) [3—6]. При этом иногда не требуется высокая скорость восстановления (восстановление старой смазанной фотографии, изображения астрономического объекта). Если требуется восстановление в режиме реального времени (восстановление опознавательных знаков на самолете-нарушителе границы, повышение качества теле- или кинокадров, томограмм динамической работы органов и др.) [4, 5], то для решения ИУ необходимо использовать быстрые алгоритмы.

В настоящей работе представлены два модифицированных быстрых алгоритма и выполняется их сравнение с существующими алгоритмами.

Математическое описание задачи восстановления смазанного изображения. Предположим, что изображение некоторого объекта искажено прямолинейным равномерным горизонтальным *смазом* (в результате сдвига фотоаппарата, самого объекта и т.д.). Обратная задача восстановления смазанного изображения [1—5] сводится к решению (в каждой у-й строке изображения) некорректного одномерного интегрального уравнения Фредгольма I рода типа свертки:

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

где

$$h(x) = \begin{cases} 1/\Delta, & -\Delta \leq x \leq 0, \\ 0, & \text{иначе,} \end{cases} \quad (2)$$

— ФРТ, обычно пространственно-инвариантная (разностная), w и g — распределение интенсивности по неискаженному (искомому) и искаженному (измеренному) изображению соот-

ветственно, Δ — величина смаза, δg — помеха. В (1) ось x направлена вдоль смаза, а y — перпендикулярно смазу и играет роль параметра. Фактически (1) есть множество одномерных интегральных уравнений. Пространственная инвариантность функции h имеет место в случае, когда величина Δ не зависит от x (но может зависеть от y , в этом, довольно редком, случае $\Delta = \Delta_y$, $h = h_y$).

Если значение Δ зависит от x (пример: смазы различны на снимке, сделанном неподвижным фотоаппаратом, где представлены легкоатлеты, бегущие с разной скоростью), то уравнение задачи о смазывании изображения будет иметь вид:

$$\int_a^b h(x, \xi) w_y(\xi) d\xi = g_y(x) + \delta g, \quad c \leq x \leq d, \quad (3)$$

где ФРТ равна (ср. с (2))

$$h(x, \xi) = \begin{cases} 1/\Delta_x, & -\Delta_x \leq x - \xi \leq 0 \text{ или } x \leq \xi \leq x + \Delta_x, \\ 0, & \text{иначе.} \end{cases} \quad (4)$$

Как в случае уравнения (1), так и (3), необходимо решить столько самостоятельных (не образующих систему) одномерных уравнений, сколько строк содержит изображение.

Решение ИУ (1) и (3). Чтобы решение $w_y(\xi)$ было устойчивым, воспользуемся методом регуляризации Тихонова. Применительно к ИУ (1) решение методом преобразования Фурье с регуляризацией Тихонова имеет вид [1, 2, 4—6]:

$$w_{\alpha y}(\xi) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{H^*(\omega) G_y(\omega)}{|H(\omega)|^2 + \alpha \omega^{2p}} e^{-i\omega \xi} d\omega, \quad (5)$$

где

$$G_y(\omega) = \int_{-\infty}^{\infty} g_y(x) e^{i\omega x} dx, \quad H(\omega) = \int_{-\infty}^{\infty} h(x) e^{i\omega x} dx \quad (6)$$

— преобразования Фурье от $g_y(x)$ и $h(x)$ соответственно, $\alpha > 0$ — параметр регуляризации, а $p \geq 0$ — порядок регуляризации (обычно 1 или 2). При компьютерной реализации формул (5) и (6) все непрерывные преобразования Фурье заменяются на дискретные, а также быстрые ПФ (БПФ) [1, 4, 6]. Число отсчетов по x и по ω полагаем одинаковым и равным n .

Для решения ИУ (3) и (1) применим метод конечных сумм (квадратур) с регуляризацией Тихонова, согласно которому интеграл при каждом x заменяется конечной суммой с целым шагом дискретизации (в пикселах) и получается система линейных алгебраических уравнений (СЛАУ) при некотором фиксированном y :

$$A w_y = g_y, \quad (7)$$

где A — $(n \times n)$ -матрица, связанная с ФРТ (4): $A_{ik} = h(x_i, \xi_k)$, причем $x_i = \xi_i = 1, 2, \dots, n$ — дискретные целочисленные отсчеты по пикселям вдоль y -й строки (n — число столбцов в изображении). Полагаем, что векторы w_y и g_y имеют размер (длину) n (в [5, с. 89—90] рассмотрены другие варианты размеров A , w_y и g_y).

Согласно методу регуляризации Тихонова вместо неустойчивой СЛАУ (7) решается:

$$(\alpha E + A^T A) w_{\alpha y} = A^T g_y, \quad (8)$$

где E — единичная $(n \times n)$ -матрица, таким образом

$$w_{\alpha y} = (\alpha E + A^T A)^{-1} A^T g_y. \quad (9)$$

Способы выбора параметра регуляризации α приведены, например, в [1, 3, 6, 7], в настоящей работе они подробно не рассматриваются. Ниже рассмотрение ведется при одном значении α (выбранном некоторым способом).

Быстрые алгоритмы вычисления решения (5). Запишем уравнение (5) в виде

$$w_{\alpha y}(\xi) = \frac{1}{2\pi} \int_{-\infty}^{\infty} Q_{\alpha}(\omega) G_y(\omega) e^{-i\omega\xi} d\omega, \quad (10)$$

где

$$Q_{\alpha}(\omega) = \frac{H^*(\omega)}{|H(\omega)|^2 + \alpha\omega^{2p}}. \quad (11)$$

Как правило, уравнения (5), (6), (10), (11) в дискретном виде решаются с использованием алгоритма БПФ [1], считающегося самым быстрым. Оценим, сколько требуется операций умножения/деления для вычисления решения (6), (10), (11) в дискретном виде с использованием БПФ (операции сложения и сопряжения не учитываем). Полагаем, что n — целая степень числа 2.

1. Один раз для всех строк вычисляем: $H(\omega)$ (потребуется $n \log_2 n$ операций), $|H(\omega)|^2$ (n операций), $\alpha\omega^{2p}$ ($2n$ операций при $p=1$) и $H^*(\omega)/|H(\omega)|^2 + \alpha\omega^{2p}$ (n операций) — всего $n \log_2 n + 4n$ операций для вычисления $Q_{\alpha}(\omega)$.

2. На вычисление $G_y(\omega)$ при некотором y потребуется $n \log_2 n$ операций.

3. На умножение $Q_{\alpha}(\omega)$ на $G_y(\omega)$ при некотором y потребуется n операций.

4. На вычисление $w_{\alpha y}(\xi)$ при некотором y потребуется $n \log_2 n$ операций.

5. В результате на вычисление $w_{\alpha}(\xi)$ для всех m строк потребуется $n(\log_2 n + 4) + n(2 \log_2 n + 1)m \approx 2mn \log_2 n$ операций.

Если $m = n$, то нужно $\approx 2n^2 \log_2 n$ операций для вычисления $w_{\alpha}(\xi)$. Таким образом, для $H(\omega)$, $Q_{\alpha}(\omega)$, $G_y(\omega)$, $Q_{\alpha}(\omega)G_y(\omega)$ и $w_{\alpha y}(\xi)$ требуется по n ячеек (всего $5n$) памяти, при некотором y , а для всех m строк потребуется $2n + 3mn \approx 3mn$ ячеек памяти. Если $m = n$, то нужно $3n^2$ ячеек.

Предлагаем другой *быстрый алгоритм*, основанный также на методе ПФ с регуляризацией (см. (5), (6)). Изменив порядок интегрирования (ср. [6, с. 261]); решение (5) можно представить в виде

$$w_{\alpha y}(\xi) = \int_{-\infty}^{\infty} \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{H^*(\omega)}{|H(\omega)|^2 + \alpha\omega^{2p}} e^{-i\omega(\xi-x)} d\omega \right] g_y(x) dx \quad (12)$$

или

$$w_{\alpha y}(\xi) = \int_{-\infty}^{\infty} R_{\alpha}(\xi-x) g_y(x) dx, \quad (13)$$

где

$$R_{\alpha}(\xi) = \frac{1}{2\pi} \int_{-\infty}^{\infty} Q_{\alpha}(\omega) e^{-i\omega\xi} d\omega. \quad (14)$$

Суть предлагаемого алгоритма состоит в том, что заранее может быть вычислен „заготовленный“ вектор R_{α} (строка теплоцевой матрицы при дискретизации) и восстановление

изображения сведется к вычислению $w_{\alpha y}(\xi)$ для каждой y -строки путем умножения R_α на g_y . Дискретизируем задачу (12)—(14). Для этого введем вдоль каждой y -й строки дискретные целочисленные отсчеты по пикселям $\xi_i = x_i = 1, 2, \dots, n$. Заменяв в (13) интеграл конечной суммой, получим:

$$w_{\alpha y}(\xi_i) = \sum_{k=1}^n R_\alpha(\xi_i - x_k) g_y(x_k). \quad (15)$$

Вычисление (15) при некотором y потребует n^2 операций умножения, а для вычисления всего изображения потребуется mn^2 . При этом не учитываем затраты на вычисление R_α , так как эти вычисления могут быть сделаны заранее. Если $m = n$, то потребуется n^3 умножений, а предыдущий алгоритм (10) (с использованием БПФ) требует $2n^2 \log_2 n$, т.е. алгоритм (15) требует больше операций умножения, чем алгоритм (10).

Что касается требуемой памяти, то алгоритм (15) требует $2n$ ячеек для R_α , а также по mn ячеек для g и для w_α , т.е. всего $\approx 2mn$ ячеек. Если $m = n$, то требуется $2n^2$ ячеек, т.е. чуть меньше, чем для алгоритма (10).

Хотя алгоритм (15) медленнее алгоритма (10), тем не менее, например, в системе MatLab чрезвычайно быстро выполняется операция умножения матрицы R_α на вектор g_y . Кроме того, в алгоритме БПФ обычно не учитывают время на вычисление экспонент („поворачивающих коэффициентов“). Если это учитывать, а алгоритм (15) реализовывать на „матричном“ языке или системе (типа MatLab), то эти два алгоритма могут сблизиться по скорости. Кроме того, алгоритм (15) требует меньше памяти и он проще алгоритма (10).

Рассмотрим также алгоритмы, основанные на соотношениях (3), (4), (7)—(9). Если решать СЛАУ (8), то на однократное вычисление матрицы $\alpha E + A^T A$ потребуется $\approx n^3$ операций умножения, на вычисление $A^T g_y$ — n^2 , а на решение СЛАУ (8) методом Холецкого (он наиболее подходит для решения СЛАУ с положительно определенной и симметричной матрицей $\alpha E + A^T A$ [8, с. 22]) потребуется $\approx n^2$. В результате на получение всего изображения w_α потребуется $n^3 + m(n^2 + n^2) = (2m + n)n^2$ операций.

Если $m = n$, то потребуется $3n^3$. Для матрицы $\alpha E + A^T A$ требуется n^2 ячеек памяти, для вектора $A^T g_y$ — n , а для решения $w_{\alpha y}$ — n , всего $n(2m + n)$ ячеек. Если $m = n$, то требуется $3n^2$ ячеек памяти.

Если искомое решение получать согласно (9), то на вычисление матрицы $\alpha E + A^T A$ потребуется n^3 операций умножения, на вычисление обратной матрицы $(\alpha E + A^T A)^{-1}$ методом Холецкого [8, с. 20—22] — $n^3/2$, на умножение $(\alpha E + A^T A)^{-1}$ на A^T — n^3 , т.е. на вычисление матрицы

$$B_\alpha = (\alpha E + A^T A)^{-1} A^T \quad (16)$$

потребуется $2n^3 + n^3/2 = 2,5n^3$. Далее, на умножение $(n \times n)$ -матрицы B_α на g_y потребуется n^2 операций, а на все строки — mn^2 . В результате на вычисление w_α согласно (9) потребуется $2,5n^3 + mn^2 = (m + 2,5n)n^2$ операций, если $m = n$, то — $3,5n^3$.

Требуемая память: если матрицу $(\alpha E + A^T A)^{-1}$ сохранять в тех же ячейках памяти $\alpha E + A^T A$, то потребуется $n(2m + n)$ ячеек; если $m = n$, то — $3n^2$.

Представим второй алгоритм, основанный на соотношении (9), записанном в виде (для каждой y -й строки изображения, ср. [5, с. 137, 6, с. 248, 251]):

$$w_{\alpha y} = B_{\alpha} g_y, \quad (17)$$

Характерная особенность решения (17) состоит в том, что матрица B_{α} может быть рассчитана заранее (несущественно, каким методом), и восстановление изображения сведется к умножению „заготовленной“ матрицы B_{α} на каждую y -ю строку изображения g . Для этого с учетом всех m строк изображения потребуется mn^2 операций. Требуемая память: для матрицы B_{α} потребуется n^2 ячеек, для g и для w_{α} — по mn , т.е. всего $n^2 + 2mn = n(2m + n)$. Если $m = n$, то потребуется $3n^2$ ячеек.

Видно, что алгоритм (17) требует такого же объема памяти, как и (8) и (9), но он требует в 3 и 3,5 раза соответственно меньше операций умножения, т.е. в 3—3,5 раза быстрее, чем (8)—(9). Кроме того, он проще в программировании.

Техническая реализация алгоритмов. Если, например, $m = n = 400$, то алгоритм (17) потребует $n^3 = 64$ млн операций умножения. Если скорость компьютера ~ 1 млрд оп/с, то восстановление изображения займет $\sim 0,1$ с. Можно время реализации алгоритма еще сократить, если использовать параллельные вычисления по строкам.

Матрица B_{α} обычно одинакова для всех y -х строк изображения (если сдвиг Δ не зависит от y), но зависит от величины смаза Δ и параметра регуляризации α . Величины Δ и α априори неизвестны. Предлагается следующий быстрый способ оценки Δ и α , который можно назвать *визуальной оценкой изображений*. Нужно заранее заготовить несколько матриц B_{α} для ряда типичных значений Δ и α . Если речь идет, например, о самолете-нарушителе границы, то служба первичного обнаружения цели должна оценить скорость, направление движения



Рис. 1

(рис. 1), а после восстановления алгоритмом „заготовленной“ матрицы, согласно (17), на самолете стали видны опознавательные знаки. Восстановление одного кадра на рис. 2 потребовало < 1 с машинного времени.

и высоте полета самолета и по этим данным служба обработки изображения сможет оценить Δ и выбрать из памяти несколько матриц B_{α} с некоторыми типичными значениями Δ и α .

В качестве примера на рис. 1 представлено смазанное изображение самолета ($m = 510$, $n = 640$, $\Delta \approx 20$). На несколько экранов могут быть выданы несколько восстановленных изображений w_{α} , и опытные диспетчеры выберут значения Δ и α (рис. 2) Видно, что при $\Delta = 20$, $\alpha = 10^{-4}$ получается вполне удовлетворительное восстановление изображения самолета: из-за смаза на самолете не были видны опознавательные знаки

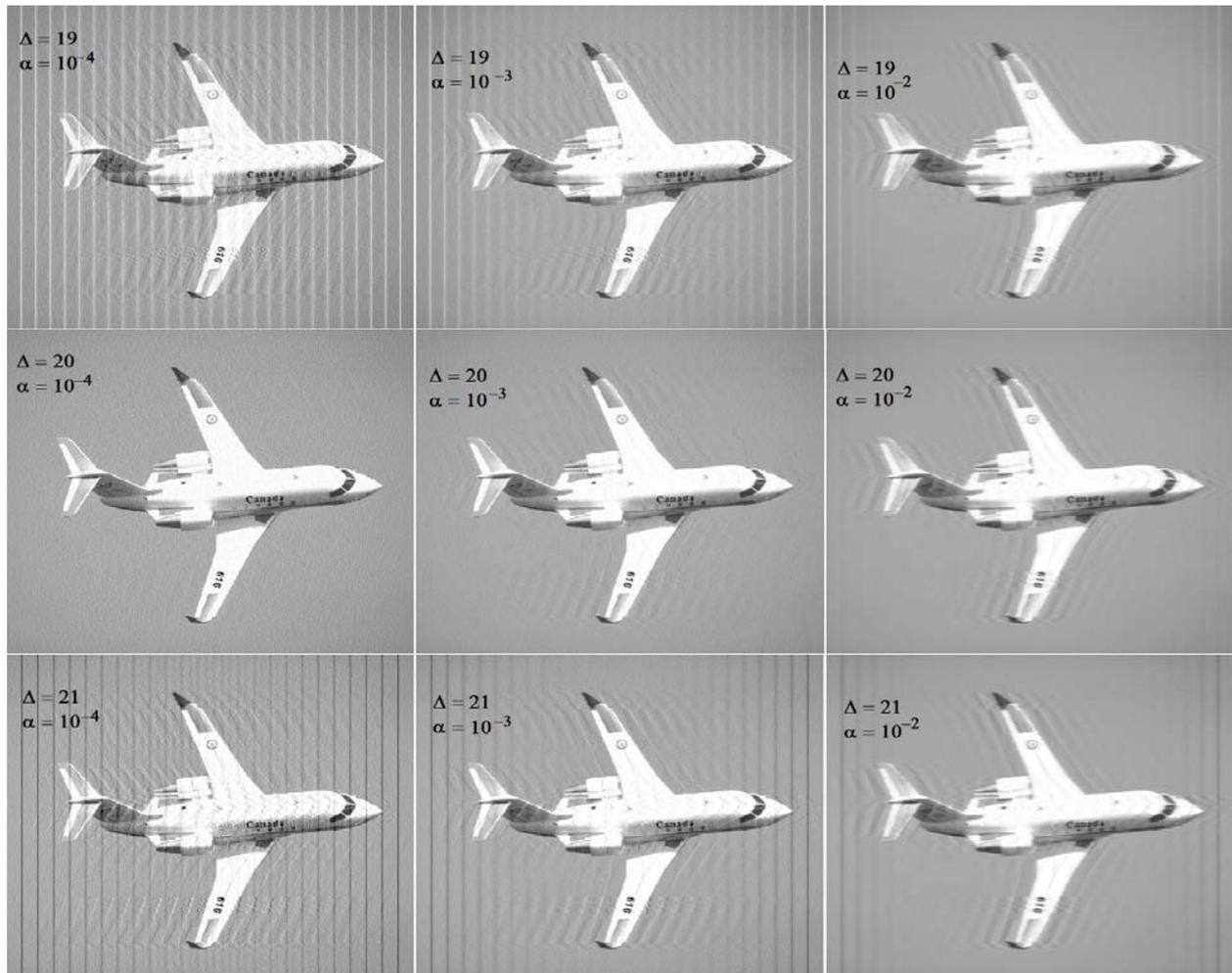


Рис. 2

Выводы. Предложенные быстрые алгоритмы могут быть использованы для восстановления смазанных изображений движущихся быстро целей (самолетов, автомобилей), для коррекции быстропротекающих процессов (смазанный кровоток в сосудах в томографии), для повышения качества теле- или кинокадров и т.д.

Работа выполнена при поддержке РФФИ (грант № 13-08-00442а).

СПИСОК ЛИТЕРАТУРЫ

1. Арефьева М. В., Сысов А. Ф. Быстрые регуляризирующие алгоритмы цифрового восстановления изображений // Вычислит. методы и программирование. 1983. Вып. 39. С. 40—55.
2. Тихонов А. Н., Гончарский А. В., Степанов В. В. Обратные задачи обработки фотоизображений // Некорректные задачи естествознания. М.: Изд-во МГУ, 1987. С. 185—195.
3. Грузман И. С., Киричук В. С., Косых В. П., Перетягин Г. И., Спектор А. А. Цифровая обработка изображений в информационных системах. Новосибирск: Изд-во НГТУ, 2002. 352 с.
4. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2006. 1072 с.
5. Сизиков В. С. Обратные прикладные задачи и MatLab. СПб: Лань, 2011. 256 с.
6. Верлань А. Ф., Сизиков В. С. Интегральные уравнения: методы, алгоритмы, программы. Киев: Наук. думка, 1986. 544 с.
7. Воскобойников Ю. Е., Мухина И. Н. Локальный регуляризирующий алгоритм восстановления контрастных сигналов и изображений // Автотометрия. 2000. № 3. С. 45—53.

8. Уилкинсон, Райни. Справочник алгоритмов на языке АЛГОЛ. Линейная алгебра. М.: Машиностроение, 1976. 389 с.

Сведения об авторах

Валерий Сергеевич Сизиков

— д-р техн. наук, профессор; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра измерительных технологий и компьютерной томографии; E-mail: sizikov2000@mail.ru

Константин Александрович Кирьянов

— аспирант; Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики, кафедра измерительных технологий и компьютерной томографии; E-mail: kiryancon@front.ru

Роман Алексеевич Экземпляров

— аспирант; Санкт-Петербургский государственный политехнический университет; E-mail: raxe@yandex.ru

Рекомендована кафедрой
измерительных технологий
и компьютерной томографии

Поступила в редакцию
13.04.13 г.

УДК 53.082.5

Г. Д. ФЕФИЛОВ

ГЕОМЕТРИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СИГНАЛА, ОСНОВАННОЕ НА ПОНЯТИИ О ФАЗОВОМ ПРОСТРАНСТВЕ, В ЛАЗЕРНОЙ ДИФРАКТОМЕТРИИ МИКРООБЪЕКТОВ

Проанализированы фазовые изображения измерительных сигналов, используемых в дифрактометрии. Предложена методика исключения избыточности сигнала, позволяющая представить параметры сигнала параметрами его фазового изображения. Такой подход позволяет получить новый информативный параметр, однозначно связанный с контролируемым размером микрообъекта.

Ключевые слова: лазерная дифрактометрия микрообъектов, информативный параметр сигнала.

В теории информации и связи сигнал традиционно представляется математической моделью в виде функции пространства или времени, характеризующей параметры исследуемого сигнала и их изменение. Также при обработке данных и выделении полезной информации широко используется описание сигналов функциями частоты. При этом любой сложный по форме сигнал представляется в виде суммы гармонических колебаний, что позволяет извлекать такую информацию об сигнале, которую трудно получить на основе непосредственного анализа его в пространственной или временной области.

При решении обратной дифракционной задачи — определении размера микрообъекта по его дифракционной картине — для выделения измерительной информации обычно используются модели сигнала, описывающие распределение интенсивности в дифракционной картине Фраунгофера в пространственной или временной области. Также используются модели в виде пространственного или временного фурье-образа сигнала, получаемого в результате амплитудной фильтрации дифракционной картины. Из измерительного сигнала выделяется информативный параметр, однозначно связанный с линейным размером объекта дифракции. Информативные параметры — основа синтеза дифракционных методик измерения, они определяют основные характеристики дифрактометров, созданных на их основе.