

ИСПОЛЬЗОВАНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ РЕШЕНИЯ  
ЗАДАЧИ ВОССТАНОВЛЕНИЯ ИЗОБРАЖЕНИЙ© 2022 г. А. О. Трубаков<sup>a,\*</sup> (ORCID: 0000-0003-0058-1215),Н. В. Медведков<sup>a,\*\*</sup> (ORCID: 0000-0002-3020-5248)<sup>a</sup> Брянский государственный технический университет,  
241035 бульвар 50 лет Октября, д. 7, Брянск, Россия

\*E-mail: trubakovao@gmail.com

\*\*E-mail: medvedkovnikita@mail.ru

Поступила в редакцию 14.12.2021 г.

После доработки 11.01.2022 г.

Принята к публикации 16.01.2022 г.

Условия фото- и видеосъемки практически никогда не бывают приближены к идеальным. Даже в относительно благоприятных внешних условиях (хорошее освещение, рассеянный свет) полученное изображение может быть испорчено движением камеры во время экспозиции или плохим фокусом. Особенно актуальна данная проблема для сфер, в которых повторно запечатлеть изображение может быть очень дорого, или вовсе невозможно.

На сегодняшний день предложено множество методов для решения этой задачи, но ни один из них не решает ее в полной мере. Существующие подходы можно разделить на два класса: ручное и автоматическое восстановление. Для ручного восстановления необходимо обладать информацией о факторах, повлиявших на искажение, а также их параметрах. На практике эти параметры практически всегда бывают неизвестны. Поэтому наибольший интерес и актуальность представляет задача автоматического восстановления искаженного изображения (так называемая “слепая” деконволюция).

Процесс решения задачи в методах “слепой” деконволюции представляет собой оптимизацию некоторой функции. В данной работе представлен обзор существующих методов в области восстановления искаженных изображений и предложен вариант восстановления на основе оптимизационной модели с использованием генетических алгоритмов со специфическими операторами рекомбинации. В существующих на данный момент методах “слепой” деконволюции в качестве оптимизационной модели чаще всего используются градиентные методы. Однако, данным методам свойственен недостаток — такие методы подвержены попаданию в локальные экстремумы. Более устойчивыми к этому являются генетические алгоритмы. Несмотря на это достоинство, в области “слепой” деконволюции существует достаточно мало исследований, посвященных использованию генетических алгоритмов в качестве оптимизационной модели. Результаты исследования и особенности применения генетических алгоритмов приведены в конце данной статьи.

DOI: 10.31857/S0132347422030116

## 1. МОДЕЛЬ ИСКАЖЕНИЯ ИЗОБРАЖЕНИЯ

Будем считать изображение идеальным, если оно было получено в идеальных условиях по освещению с использованием идеальных регистрирующих приборов, не вносящих никаких искажений. На самом деле таких изображений не бывает (если только это не смоделированное изображение компьютерной графики). Искажения на реальных изображениях можно описать следующей моделью [1, 2]:

$$g(x, y) = f(x, y) \times h(x, y) + n(x, y), \quad (1.1)$$

где:  $f(x, y)$  — идеальное изображение;  $h(x, y)$  — искажающая функция, описывающая внешние влияния на процесс регистрации изображения;

$n(x, y)$  — аддитивная составляющая, прибавленная к изображению в процессе регистрации и оцифровки (шумовая составляющая);  $\times$  — операция свертки;  $g(x, y)$  — полученное в результате изображение.

Таким образом, негативными компонентами, проявившимися в процессе регистрации изображения, являются искажающая функция  $h(x, y)$  и шумовая составляющая  $n(x, y)$ . Избавившись от них, можно получить изображение, близкое к идеалу. В реальных условиях обеспечить идеальность процесса съемки нельзя, поэтому встает задача разработки такого математического аппарата, который по известному изображению  $g(x, y)$



Рис. 1. Слева – искажающая функция, справа – искаженное изображение.

подберет такой оператор, который нейтрализует негативный эффект от  $h(x, y)$  и  $n(x, y)$ . Именно этот процесс называется восстановлением искаженных изображений.

В математическом плане искажающая функция – это ядро фильтра в пространственной области. Соответственно, процесс искажения изображения – это обычная фильтрация с ядром, которое напрямую зависит от вида искажающей функции. Если искажающая функция имеет вид прямой линии, то изображение будет прямолинейно смазано (см. рис. 1). Такое поведение часто проявляется при движении или дрожании регистрирующего прибора и является наиболее частым примером искажений в реальных ситуациях.

Выше было показано, что помимо искажающего ядра свертки к изображению добавляется аддитивная составляющая (шум), вносимая фотосенсорами и электроникой устройств вследствие несовершенства технологий, а также фотонной природы света. Стоит отметить, что накладываемый шум никак не зависит от изображения и никак с ним не коррелируется и в большинстве случаев является случайным.

Восстановление растровых изображений – это процесс получения некоторого приближения четкого идеального изображения из имеющегося искаженного. Математически модель восстановления можно описать следующим образом:

$$\hat{f}(x, y) = R(g(x, y)), \quad (1.2)$$

где  $\hat{f}(x, y)$  – приближение идеального изображения, полученное в результате восстановления;  $R(I(x, y))$  – восстанавливающий оператор;  $g(x, y)$  – искаженное изображение.

На данный момент существует большое количество исследований, посвященных решению задачи восстановления изображений. Несмотря на

это, пока не найдено такое решение, которое являлось бы полным решением задачи [3]. Среди методов решения задачи восстановления растровых изображений можно выделить три основных подхода:

- “не слепая” деконволюция (ручное восстановление);
- “слепая” деконволюция (автоматическое восстановление);
- глубокое обучение (автоматическое восстановление).

Ручное восстановление представляет собой применение одного из восстанавливающих фильтров или алгоритмов, в которых входными данными являются не только искаженные изображения, но также и искажающая функция (фильтр), и специфичные для фильтра параметры [4]. Ручное восстановление редко применяется на практике, так как искажающая функция или ее параметры практически всегда являются неизвестными величинами, а их подбор является достаточно трудоемким процессом и требует знаний в предметной области. Однако классические методы ручного восстановления широко применяются в методах “слепой” деконволюции как составляющие итерационного этапа поиска приближения.

Автоматическое восстановление без априорных знаний является крайне сложной задачей, так как в процессе решения неизвестными являются не только четкое изображение, но и искажающая функция [5, 6]. При этом нужно понимать, что существует бесчисленное множество пар искажающей функции и четкого изображения, которые при свертке давали бы наблюдаемое искаженное изображение. Поэтому точного и однозначного решения задачи в большинстве случаев не существует, но можно строить модели, которые будут выдавать такой результат, который бы как можно ближе был к оригиналу и удовлетворял потребностям пользователя.

Автоматическое восстановление изображений на данный момент представлено двумя классами методов: “слепая” деконволюция и глубокое обучение. Первый класс методов представляет собой более точные методы, для которых характерно применение сложных математических моделей и высокая вычислительная сложность. Для второго класса методов характерна высокая зависимость получаемого результата от обучающей выборки. В последнее время наблюдается бурное развитие глубокого обучения и нейросетей. Однако, данные методы могут давать неточные результаты – “восстановить” те элементы на изображениях, которых не было в оригинальном изображении [7]. Методы “слепой” деконволюции дают более точные результаты, что делает их наиболее предпочтительным выбором, особенно для сфер, в которых точность и правдоподобие явля-

ется очень важным фактором. В следующем разделе будут более подробно рассмотрены методы восстановления изображений.

## 2. ОБЗОР МЕТОДОВ ВОССТАНОВЛЕНИЯ РАЗМЫТЫХ ИЗОБРАЖЕНИЙ

**Ручное восстановление.** Методы ручного восстановления (“не слепой” деконволюции) представляют собой восстанавливающие фильтры и алгоритмы, которые в качестве входных параметров принимают как размытое изображение, так и модель искажающей функции, заданной вручную со всеми ее параметрами. Методы “не слепой” деконволюции подразумевают, что искажающая функция известна полностью и вся задача сводится к применению одного из восстанавливающих фильтров или алгоритмов [8].

Простая инверсная фильтрация, которую можно получить из уравнения модели искажения (1.1), не учитывает аддитивный шум, что может привести к негативным последствиям. Поэтому стоит использовать другие фильтры, которые дают более корректные результаты. Одним из них является фильтр Винера [9, 10], в котором рассчитывается четкое изображение, дающее минимальное среднеквадратическое отклонение между исходным и восстановленным изображением:

$$F(u, v) = \left( \frac{1}{H(u, v) \left( |H(u, v)|^2 + \frac{S_n(u, v)}{S_f(u, v)} \right)} \right) G(u, v), \quad (2.1)$$

где  $F(u, v)$  – приближение четкого изображения;  $H(u, v)$  – искажающая функция;  $G(u, v)$  – результат искажения (размытое или смазанное изображение);  $S_n(u, v)$  – спектральная плотность шума;  $S_f(u, v)$  – спектральная плотность исходного изображения;  $\frac{S_n(u, v)}{S_f(u, v)}$  – редко бывает известно, поэтому данное отношение заменяют на некоторую константу  $K$ .

Еще одним часто используемым методом является сглаживающая фильтрация методом наименьших квадратов со связью. Этот метод известен как “фильтрация по Тихонову” или “Тихоновская регуляризация” [9, 10]. В этом фильтре задача восстановления представляется в матричной форме и происходит оптимизация модели. Решение принимает вид:

$$F(u, v) = \left( \frac{H(u, v)}{|H(u, v)|^2 + y|P(u, v)|^2} \right) G(u, v), \quad (2.2)$$

где  $y$  – параметр регуляризации, а  $P(u, v)$  – Фурье-преобразование оператора Лапласа (матрицы  $3 \times 3$ ).

Еще одним методом “слепой” деконволюции является метод Ричардсона–Люси. Этот метод использует подход максимального правдоподобия и является нелинейным и итерационным [9]. Для его работы необходимо задать еще один параметр – число итераций. В отличие от первых двух методов, которые используют образы Фурье [10] (работают преимущественно в частотной области), данный метод обрабатывает изображение в пространственной области:

$$f_{k+1}(x, y) = f_k(x, y) \left( h(-x, -y) \frac{g(x, y)}{h(x, y) \times f_k(x, y)} \right), \quad (2.3)$$

где  $f_k$  – приближение четкого изображения на итерации номер  $k$ ;  $h(x, y)$  – искажающая функция;  $k$  – номер итерации;  $g(x, y)$  – результат искажения.

Метод Ричардсона–Люси широко используется в программах для обработки астрономических фотографий. Недостатком данного метода является очень большая вычислительная сложность.

**“Слепая” деконволюция.** Методы ручного восстановления позволяют восстанавливать изображение с достаточно хорошим результатом, но для этого требуется точно описать функцию искажения. Однако, в большинстве реальных ситуаций искажающую функцию либо очень сложно описать, либо она вовсе неизвестна. В таких ситуациях задача “слепой” деконволюции превращается в задачу “слепой” деконволюции, цель которой – поиск приближения четкого изображения и искажающей функции без наличия информации об искажающей функции [5, 6]. Задача в такой постановке становится на порядок сложнее. Во-первых, возрастает количество неизвестных переменных – к неизвестному приближению четкого изображения добавляется неизвестная искажающая функция. Во-вторых, с математической точки зрения задача имеет множество решений, поскольку существует бесчисленное множество пар изображения и искажающей функции, свертка которых давала бы наблюдаемое искаженное изображение. Таким образом, данную задачу можно назвать задачей с недостаточным количеством информации для ее решения.

Для решения данной задачи было предложено множество методов, использующих разные подходы. Некоторые методы поиска искажающей функции включают восстановление с помощью методов фильтрации, например, Giannakis и Heath [11] разработали последовательность фильтров с конечной импульсной характеристикой для непосредственного получения приближения четкого изображения и последовательность фильтров с конечной импульсной характеристикой для определения функции искажения. Hosseini и Plataniotis [12] также представили метод фильтрации с использованием фильтров конечной импульсной харак-

теристики. Также были представлены множество методов, использующих метод Люси–Ричардсона [13], подход Total Variation [14], метод максимального правдоподобия [15] и др. Достаточным распространением является пирамидальный подход, при котором строится пирамида разных разрешений искаженного изображения, и неизвестная искажающая функция восстанавливается постепенно — она уточняется в процессе оптимизации некоторой целевой функции на каждом уровне пирамиды от самого низкого разрешения (первого уровня пирамиды) до самого высокого разрешения (последнего уровня пирамиды) [1, 2, 6, 16].

*Методы, основанные на теории линейных моделей.* В теории линейных моделей искажение изображения понимается как процесс перехода четких границ в размытые. Подразумевается, что данный процесс линеен и размытие является самой весомой операцией в данном процессе. Также подразумевается, что четкое и размытое изображение — черно-белые и являются цифровыми. В целях математической реализации линейной модели, также подразумевается, что четкое и искаженное изображения — матрицы размером  $m \times n$  и искажение (размытие) строк и столбцов изображения происходит отдельно. Для удобства последующего изложения, в данном разделе будут использоваться следующие обозначения:  $F$  — матрица размера  $m \times n$ , представляющая четкое неискаженное изображение;  $G$  — матрица размера  $m \times n$ , представляющая размытое изображение;  $N$  — матрица размера  $m \times n$ , представляющая аддитивный шум;  $A^{-1}$  — матрица, обратная матрице  $A$ ;  $A^T$  — транспонированная матрица  $A$ ;  $A_c$  — матрица размера  $n \times n$ , представляющая размытие по столбцам;  $A_r$  — матрица размера  $m \times m$ , представляющая размытие по строкам.

Отношение между четким и размытым изображением можно представить следующей формулой:

$$G = A_c F A_r^T, \quad (2.4)$$

где  $A_c F$  — размытие каждого столбца в четком изображении  $F$ ;  $F A_r^T$  — размытие каждой строки в четком изображении  $F$ .

Для получения четкого изображения  $F$  формулу выше нужно преобразовать к следующему виду:

$$F = A_c^{-1} G (A_r^T)^{-1}. \quad (2.5)$$

Проблема состоит в том, что данное уравнение представляет собой обычную инверсную фильтрацию, и получаемое таким образом приближение  $F$  не отображает действительно четкое изображение, так как в данном подходе не учитывается шум, накладываемый на изображение.

Для того, чтобы избежать данной проблемы в исходное уравнение (2.4) включают аддитивный

шум, таким образом, изначальное уравнение (2.4) выглядит следующим образом:

$$G = A_c F A_r^T + N. \quad (2.6)$$

Для получения приближения четкого изображения  $F$  из данной формулы пользуются следующим подходом:

$$F = A_c^{-1} G (A_r^T)^{-1} - A_c^{-1} N (A_r^T)^{-1}. \quad (2.7)$$

Faramarzi и др. [17] разработали обобщенный метод, основанный на теории линейных моделей, который решает как задачу слепой деконволюции, так и задачу супер-разрешения. Задача восстановления размытых изображений может быть представлена как подзадача супер-разрешения. В данном подходе линейная модель используется для получения четких изображений с более низким разрешением из размытого изображения с более высоким разрешением.

*Методы, основанные на стохастических моделях.* Стохастические модели основаны на анализе функции плотности вероятности размытого изображения. Например, Xiao и другие [18] разработали алгоритм слепой деконволюции “Stochastic Random Walk Optimization Algorithm”, который основан на простом методе случайного поиска. В данном подходе промежуточное решение обновляется за счет получаемой цепочки случайных перемещений по пикселям в размытом изображении, после чего обновленное решение проверяется на соответствие цели. После проверки решается применять ли это обновление. Этот метод итеративно выполняет вышеописанный алгоритм для поиска неизвестной функции искажения, которая используется в алгоритме “не слепой” деконволюции для получения приближения четкого изображения. В данной работе был использован алгоритм “не слепой” деконволюции из предыдущей работы [19], посвященной стохастическому подходу эффективной “не слепой” деконволюции. Особенностью данного алгоритма является его простота.

*Методы, основанные на вероятностной модели Байеса.* Два наиболее популярных в настоящий момент метода восстановления изображений основаны на Байесовском подходе оценки апостериорного распределения с использованием его отношения с функцией правдоподобия и априорным распределением. Данное отношение можно выразить формулой:

$$\begin{aligned} \rho(f, h|g) &= \frac{\rho(g|f, h)\rho(f)\rho(h)}{\rho(g)} \propto \\ &\propto \rho(g|f, h)\rho(f)\rho(h), \end{aligned} \quad (2.8)$$

где  $\rho(f, h|g)$  — апостериорное распределение изображения, которое пропорционально функции правдоподобия  $\rho(g|f, h)$  и априорным рас-

пределениям  $\rho(f)$  и  $\rho(h)$  изображения и функции искажения соответственно.

Первая группа методов, использующая вероятностную модель Байеса, использует вариационные Байесовские методы [20, 21]. Данная группа методов весьма эффективна для задачи точного поиска функции искажения для различных видов размытия таких как размытие и криволинейный смаз. Главным недостатком данных методов является большая вычислительная сложность из-за необходимости постоянного расчета интегралов.

Вторая группа методов основана на оценке апостериорного максимума. Стоит отметить, что данная группа методов является самой популярной для решения задачи “слепой” деконволюции. Оценка апостериорного максимума основана на том, что оптимальное значение логарифма функции является оптимальным значением исходной функции, поскольку логарифм является монотонной функцией. Эта модель может быть использована с любым видом априорной информации о изображении или о функции искажения на различных этапах алгоритма.

Ранее считалось, что метод оценки апостериорного максимума может использоваться для поиска только тривиальных функций искажения, однако, недавние исследования показали, что этот метод может использоваться и для поиска комплексных функций искажения большой сложности [22, 23].

Метод оценки апостериорного максимума похож на метод максимального правдоподобия, в котором поиск неизвестной информации осуществляется с использованием некоторых известных экспериментальных данных. Это выполняется путем максимизации апостериорного распределения.

Метод оценки апостериорного максимума можно использовать двумя способами. Первый способ заключается в одновременной оценке функции искажения и размытого изображения:

$$\max_{f,h} \rho(f, h | g) \propto \max_{f,h} \rho(g | f, h) \rho(f) \rho(h) \quad (2.9)$$

где:  $\max_{f,h} \rho(g | f, h)$  – максимальное правдоподобие;  $\rho(f)$  и  $\rho(h)$  – априорные распределения изображения и функции искажения соответственно.

В рамках задачи слепой деконволюции данное выражение представляется в виде задачи минимизации следующей функции:

$$\begin{aligned} (\hat{f}, \hat{h}) = \min_{f,h} & \lambda \|f \times h - g\|^2 + \\ & + \sum_i |k_{x,i}(f)|^\alpha + \sum_i |k_{y,i}(f)|^\alpha \end{aligned} \quad (2.10)$$

где  $\lambda \|f \times h - g\|^2$  представляет собой правдоподобие, которое одновременно является естественным ограничением, накладываемым на искомую искажающую функцию и приближение четкого изображения;  $|k_{x,i}(f)|^\alpha$  и  $|k_{y,i}(f)|^\alpha$  представляют собой горизонтальные и вертикальные производные пикселя  $i$  приближения четкого изображения  $f$ .

В данном выражении первое слагаемое является постоянным, а производные могут быть изменены, путем задания параметра  $\alpha$ . Разные значения данного параметра подходят для разных видов изображений лучше всего подходят значения от 0.5 до 0.8 [24]. Также стоит отметить, что с ростом разрешения изображения количество неизвестных возрастает, и для данного метода более предпочтительным является размытое изображение, это может быть объяснено тем, что размытие оказывает два противоположных эффекта на значение правдоподобности: 1) оно делает изображение более сглаженным, что уменьшает значение правдоподобия; 2) размытие уменьшает дисперсию производных, что увеличивает значение правдоподобия [3]. И, в конце концов, самым предпочтительным результатом в соответствии с выражением выше будет являться изображение, не содержащее градиентов вовсе. Данная проблема характерна даже для ситуаций, в которых параметр  $\alpha$  выбран достаточно точно для восстанавливаемого изображения, и использование производных более высокого порядка не оказывает положительного эффекта на результат [25, 26].

Второй способ заключается в поиске (оценке) только функции искажения, которая далее используется в одном из алгоритмов “не слепой” деконволюции для получения приближения четкого изображения  $f$ . Как было сказано ранее, с увеличением разрешения изображения, способ одновременного поиска функции искажения и приближения четкого изображения ведет себя некорректно. В связи с этим второй способ вместо одновременного поиска четкого изображения и функции искажения занимается только поиском последнего, пользуясь тем, что разрешение искажающей функции не меняется и достаточно мало по сравнению с разрешением изображения. В соответствии с данным подходом, функция искажения принимает следующий вид:

$$\hat{h} = \arg \min_h \rho(g | h) \rho(h) / \rho(g), \quad (2.11)$$

где

$$\rho(g | h) = \int \rho(f, g | h) df \propto \int e^{\frac{-1}{2\sigma^2} |h-g|^2 - \frac{f^2}{2\sigma^2}} df. \quad (2.12)$$

Если предположить, что шум и изображение нормально распределены с нулевым математическим

ожиданием и дисперсиями  $\delta^2$  и  $\eta^2$ :  $f = N(0, \delta^2)$  и  $n = N(0, \eta^2)$ .

Описанный алгоритм работает в частотной области, что позволяет избежать дорогостоящих с точки зрения вычислений операции свертки. Таким образом, алгоритм начинается с получения образа Фурье изображения. После этого алгоритм делится на два повторяющихся шага: шаг поиска приближения четкого изображения и шаг поиска функции искажения. На каждом из этих шагов минимизация функции выполняется с использованием расширенных Лагранжевых методов или с помощью другой оптимизационной модели, которой чаще всего является одна из вариаций метода градиентного спуска. В результате оптимизации получается функция искажения и приближение четкого изображения. Главное требование алгоритма представляется к оптимизационной модели, а именно – достижение глобального минимума, не попадание в локальные экстремумы.

Было предложено множество модификаций данного подхода, например, Perrone и Favaro [27] показали как работает простая версия алгоритма, описанного выше, и описали его детали работы, и то, почему на них стоит заострять внимание. В результате они предложили свой метод, названный алгоритмом “Прогнозируемой чередующейся минимизации” (Projected Alternating Minimization (PAM)), который учитывает множество деталей, например, предположение о функции искажения, что улучшает поиск функции искажения на поздних стадиях алгоритма. Также авторы наглядно продемонстрировали, что использование оценки апостериорного максимума для поиска функции искажения и приближения четкого изображения приводит к попаданию в локальный минимум, поэтому в предложенном ими алгоритме было решено отказаться от данной идеи.

Также была представлена еще одна модификация – алгоритм ускоренной чередующейся минимизации (Accelerated Alternating Minimization Algorithm) [28], представленный Padcharoen и др. Данная модификация использует подход, называемый методом ускоренного проксимального градиента (Accelerated Proximal Gradient (APG)), который используется для увеличения скорости сходимости обычно медленного градиентного спуска.

Также стоит отметить, что очень часто к методу оценки апостериорного максимума применяют регуляризацию – метод добавления дополнительных ограничений или априорных распределений на параметры модели в задачах с недостаточным количеством информации для решения. Наиболее частым подходом регуляризации в рамках рассматриваемой задачи является следующий:

$$\min_f = \frac{1}{2} \|f - g\|_2^2 + \frac{\lambda}{2} \Phi(f), \quad (2.13)$$

где  $f$  – приближение четкого изображения;  $g$  – размытое изображение;  $\Phi(f)$  – регуляризатор;  $\lambda$  – параметр регуляризации.

От метода к методу  $\Phi(f)$  может варьироваться, но наиболее распространенным регуляризатором в методах, основанных на оценке апостериорного максимума, является  $L_p$  – норма:

$$\Phi(f) = \left( \sum_{i,j} |f_{i,j}|^p \right)^{\frac{1}{p}}, \quad (2.14)$$

где  $i$  и  $j$  – индексы элемента матрицы.

**Глубокое обучение.** Сверточные нейронные сети нашли широкое применение в обработке изображений и компьютерном зрении. Автоматическое восстановление размытых и смазанных изображений не стало исключением. Как правило, восстановление размытых и смазанных изображений с помощью нейросетей нацелено не на поиск неизвестной функции искажения, а на улучшение визуального качества изображения. Но для справедливости стоит отметить, что существуют модели, которые ориентированы именно на поиск неизвестной искажающей функции [29]. Модели глубокого обучения для восстановления размытых и смазанных изображений широко используют в своей основе энкодеры и декодеры [30, 31] и генеративно-состязательные нейронные сети [32, 33].

Преимущество методов, основанных на глубоком обучении перед методами, основанных на деconvолуции состоит в том, что данные методы могут хорошо показывать себя при восстановлении частично размытых изображений, при условии, что подобные модели будут обучены соответствующим образом.

Нейросетевые модели очень чувствительны к обучающей выборке. В качестве примера зависимости качества обучения модели от обучающей выборки стоит рассмотреть пример восстановления с помощью обученной модели DeblurGAN, при обучении которой в обучающей выборке отсутствовали зашумленные изображения (см. рис. 2).

Так же к недостаткам стоит отнести ситуации, когда обученная модель “дорисовывает” несуществующие детали на изображении. При этом стоит отметить, что большинство таких результатов обученной модели действительно будут являться визуально более четкими, но с точки зрения пользователя некоторые результаты будут абсолютно неудовлетворительными из-за деталей, которые добавила нейросеть (см. рис. 3).

### 3. ПРЕДЛАГАЕМОЕ РЕШЕНИЕ НА ОСНОВЕ ГЕНЕТИЧЕСКОГО АЛГОРИТМА

**Допущения, сделанные при исследовании.** На данном этапе исследований было решено сделать





**Рис. 2.** Слева – размытое изображение с сильным шумом, справа – результат восстановления с помощью DeblurGAN.

ряд допущений, которые позволили сконцентрировать исследования на генетическом алгоритме, опустив некоторые сложности. Во-первых, используется идеализированная целевая функция: результат искажения сравнивается с известным эталонным четким изображением с использованием метрики SSIM [34]. Однако, как показали эксперименты, одного структурного сходства между изображениями оказывается недостаточно. В целевой функции должна присутствовать информация о том, насколько четкое изображение мы получаем. В противном случае алгоритм останавливается на функции искажения, представляющей из себя точку из нескольких пикселей. Во-вторых, используются искусственно размытые изображения без последующего наложения шума.

Сами по себе эти допущения являются весьма серьезными. Однако все ограничения касаются оптимизируемой целевой функции, и позволяют продемонстрировать работоспособность предлагаемого подхода на данном этапе работы.

**Пирамидальный подход.** Предлагаемый алгоритм основан на пирамидальном подходе, который нашел широкое применение в методах “слепой” деконволюции [1, 2, 6, 13, 16]. Пирамидальный подход заключается в постепенном уточнении ядра от самого маленького размера до максимального размера, задаваемого пользователем. Процесс поиска ядра свертки (функции искажения) начинается, как правило, с разрешения  $3 \times 3$  или  $5 \times 5$ . При этом в процессе оптимизации ядра определенного размера используется искаженное изображение не исходных размеров, а уменьшенного размера. Для данной цели строится пирамида искаженных изображений разных размеров. Каждое такое изображение соответствует определенному размеру ядра свертки. Перемещение по пирамиде дает возможность уточнять ядро на каждом шаге, а не подбирать его заново, что значительно экономит время и ресурсы при работе алгоритма.

При проектировании увеличение разрешения ядра было решено сделать плавным (на 2 по ши-



**Рис. 3.** Слева – размытое изображение, справа – результат восстановления с помощью DeblurGAN.

рине и высоте на каждый уровень пирамиды). Экспериментальная проверка показала, что резкое увеличение ядра пагубно влияет на результат оптимизации. При этом стоит отметить, что ядра на начальных уровнях пирамиды инициализируются горизонтальными линиями максимальной яркости.

**Генетический алгоритм.** Генетические алгоритмы оперируют понятием особи – точка в пространстве поиска, которой в данном случае является ядро, то есть квадратная матрица. В классической постановке с каждой координатой точки пространства поиска принято работать в закодированном виде. В данном случае особь будет представлять из себя хромосому – последовательность генов [35]. Однако, в предлагаемом алгоритме было решено отказаться от данной идеи и работать с фенотипом особи – то есть в незакодированном виде. При этом предлагается работать с незакодированными параметрами особи как с генами. В связи с этим стоит обозначить, что далее для удобства изложения особь будет называться ядром, а незакодированный параметр особи будет обозначаться пикселом. Также стоит упомянуть, что область определения яркости пикселей ядер представляет собой отрезок  $[0; 1]$ . Все обрабатываемые изображения нормализованы от 0 до 1.

**Селекция.** В качестве оператора селекции было решено использовать турнирную селекцию. Это сделано потому, что данный вид селекции позволяет регулировать разнообразие родительского пула с помощью размера турнира. Чем больше размер турнира, тем больше особей с высокой приспособленностью будет в родительском пуле. Чем меньше размер турнира – тем больше шансов у особей с низкой приспособленностью попасть в родительский пул. Это делает турнир более разнообразным. В предлагаемом алгоритме рекомендуется использовать размер турнира, равный двум.

**Скрещивание.** Для скрещивания в предлагаемом алгоритме используется классическое равномерное скрещивание без каких-либо модификаций. Было принято решение использовать данный вид скрещивания из-за особенности представления особи. Особь представляет собой не одно-

мерный массив, а двумерный. Это не просто набор значений переменных, но и еще изображение некоторой непрерывной функции. Поэтому от операторов одноточечного или двухточечного скрещивания было решено отказаться, так как “пересадка” участка изображения в другой участок изображения для данной задачи будет являться крайне неэффективной.

**Мутация.** Искомое ядро свертки, как правило, представляют из себя изображение, большая часть пикселей которого имеет яркость 0. При этом только небольшое количество пикселей в центре изображения имеет яркость больше нуля. В ходе экспериментов выяснилось, что классические операторы мутации, такие как инвертирование бита, мутация обменом, или мутация обращением [35] не производят положительного эффекта на процесс оптимизации, так как мутации в данном случае происходят по всему ядру, и количество “полезных” мутаций значительно сокращается. Это проявляется в первую очередь в изменении пикселей по краям ядра, которые в большинстве случаев должны быть равны нулю. Данная мутация уводит процесс оптимизации в сторону и увеличивает время работы алгоритма. В процессе исследования было выявлено, что если мутации происходят возле найденной на предыдущей итерации функции искажения (пикселей с высокими значениями), то количество “полезных” мутаций становится гораздо больше. Исходя из данных наблюдений, было решено производить мутации вблизи пикселей с высокими значениями. В предложенном алгоритме высоким значением является значение из отрезка  $[0.1; 1.0]$ . Таким образом, количество пикселей для мутации сильно сокращается.

Стоит также отметить, что мутации поддается не сам пиксель, а пиксель, находящийся возле него. В предложенном алгоритме данная идея реализована путем выбора случайного пикселя из квадратной области, центром которой является выбранный пиксель с высоким значением.

Так как ранее было упомянуто, что работа с особями ведется в закодированном виде, то операция инвертирования в данном случае теряет смысл. Также стоит отметить, что изменение значения на значение противоположного края отрезка области определения в данном случае не эффективно, так как большинство реальных функций искажения не представляют из себя бинарные изображения и могут иметь промежуточные значения яркости. Поэтому в качестве операций мутации в алгоритме выступает случайное положительное или отрицательное изменение значения пикселя на случайное значение из области определения (вещественное число от 0 до 1). При этом вероятность положительной мутации задается отдельно, и, как показали эксперименты, оптимальным

значением данной вероятности являются значения из отрезка  $[0.5, 0.9]$ .

**Элитизм.** В предлагаемом алгоритме рекомендуется использовать 1 элитную особь для сохранения оптимального решения. Таким образом не тратится время на повторный поиск уже найденного решения. Большое количество элитных особей при проведении исследований не сказывалось положительно на результатах поиска решений.

Также с положительной стороны себя проявил авторский подход без использования элитных особей. При этом лучшая особь из каждой популяции сохранялась в отдельный пул, и после того, как оптимизация для текущего уровня пирамиды была выполнена полностью (перед переходом на новый уровень пирамиды с большим ядром), особи заменялись на усредненный вариант лучших особей из вышеупомянутого пула. Но стоит отметить, что данный вариант проигрывает варианту с 1 элитной особью, хоть и вероятность попасть в локальный максимум в данном случае гораздо ниже.

**Алгоритм и результаты его работы.** Рассмотрим общий алгоритм оптимизации с использованием генетических алгоритмов. Блок-схема предлагаемого алгоритма представлена на рис. 4.

На приведенной выше блок-схеме шаг “ввод данных” имеет особый смысл. Под входными данными подразумевается: искаженное изображение; вероятность скрещивания; максимальный размер ядра; вероятность мутации; вероятность положительного изменения значения пикселя при мутации; количество популяций, в течение которых не происходит увеличения целевой функции (критерий останова).

Стоит отметить, что было решено увеличить количество особей в популяции при переходе на новый уровень пирамиды. В ходе экспериментов выяснилось, что с увеличением размерности ядра, данный аспект существенно влияет на результат оптимизации. Однако он имеет и обратную негативную сторону – при этом существенно снижает скорость оптимизации. Для проведения экспериментов была составлена коллекция изображений небольшого разрешения (меньше разрешения  $1000 \times 1000$  пикселей). Все изображения коллекции были искусственно искажены с помощью заранее подготовленных функций искажения.

Для инверсной фильтрации использовался фильтр Винера. В качестве целевой функции использовалась функция, представляющая собой комбинацию двух метрик: метрики, оценивающей сходство результата восстановления с оригиналом и метрики, оценивающей визуальную четкость результата. В качестве метрики сходств была использована метрика структурного сходства SSIM, а роль метрики, оценивающей то, насколько визуально четким получается изображение,



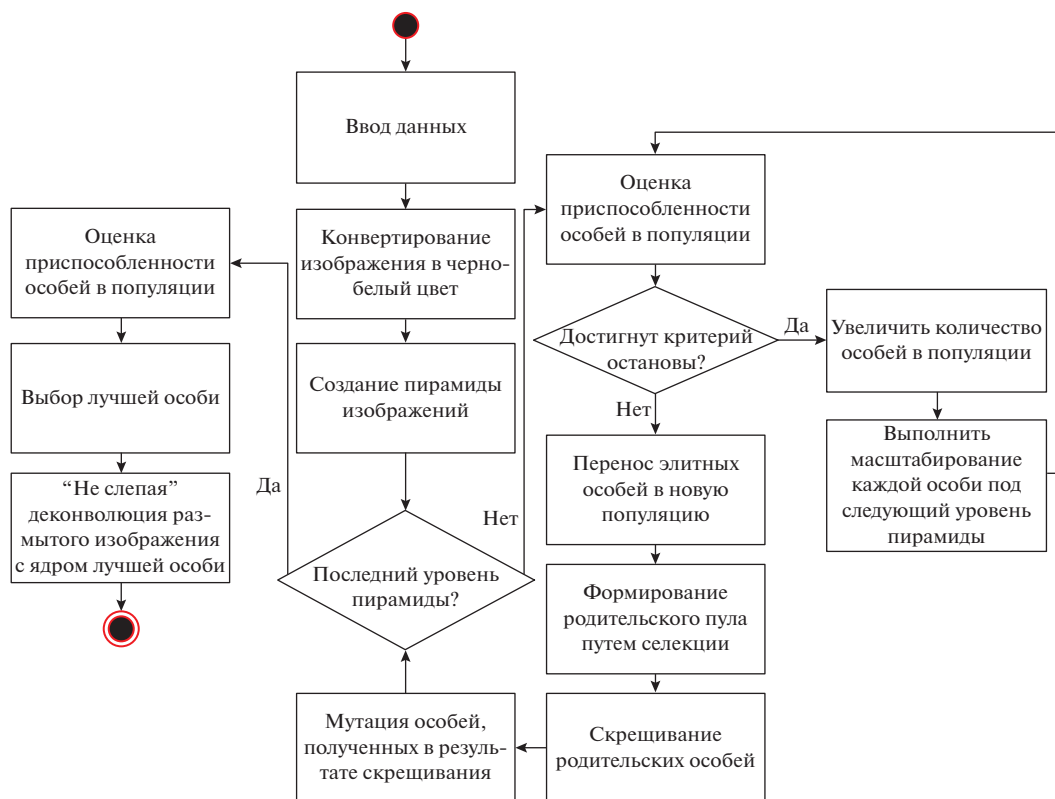


Рис. 4. Блок-схема алгоритма.

выполняла метрика, представляющая собой среднее значение образа Фурье-изображения:

$$Q_F(I) = \frac{\sum_{j=1}^n \sum_{i=1}^m DFT(I)(i, j)}{m \times n}, \quad (3.1)$$

где  $I$  – оцениваемое изображение разрешения  $m \times n$ ;  $i, j$  – координаты пиксела;  $DFT(I)$  – образ Фурье изображения  $I$ , полученный с помощью дискретного преобразования Фурье;  $g(x, y)$  – результат искажения (размытое или смазанное изображение).

Таким образом, целевую функцию можно описать следующим выражением:

$$Q(Orig, Res) = SSIM(Orig, Res) + \log(Q_F(Res)), \quad (3.2)$$

где  $Orig$  – четкое изображение;  $Res$  – результат восстановления.

Для проверки поставленных гипотез был проведен ряд экспериментов над изображениями, разными по характеру исполнения и деталей. Каждое изображение было искажено разными функциями, от простых, до более сложных. Все эксперименты выполнялись со следующими значениями параметров: вероятность скрещивания: 0.1; размер турнира: 2; максимальный размер яд-

ра: 23; вероятность мутации: 0.1; вероятность положительного изменения значения пиксела при мутации: 0.5; количество популяций, в течении которых не происходит увеличения целевой функции (критерий останова): 15.

Один из примеров исходного четкого изображения, искажающая функция, а также результат искажения и восстановления представлены на рис. 5.

Как можно заметить, найденная искажающая функция имеет очень хорошее сходство по форме с настоящей функцией искажения, используемой при подготовке исходных данных. Однако можно заметить, что найденная функция значительно ярче действительной, а также имеет отдельно стоящие пиксели. Этих результатов можно избежать, проведя в дальнейшем дополнительные эксперименты с метрикой оценки, являющейся основой для функции оптимизации.

#### 4. ВЫВОДЫ

В работе представлен обзор методов восстановления искаженных изображений и новый метод, основанный на использовании пирамиды изображений и генетических алгоритмов для задачи оптимизации и поиска ядра. Как показали эксперименты, описанный вариант дает доста-

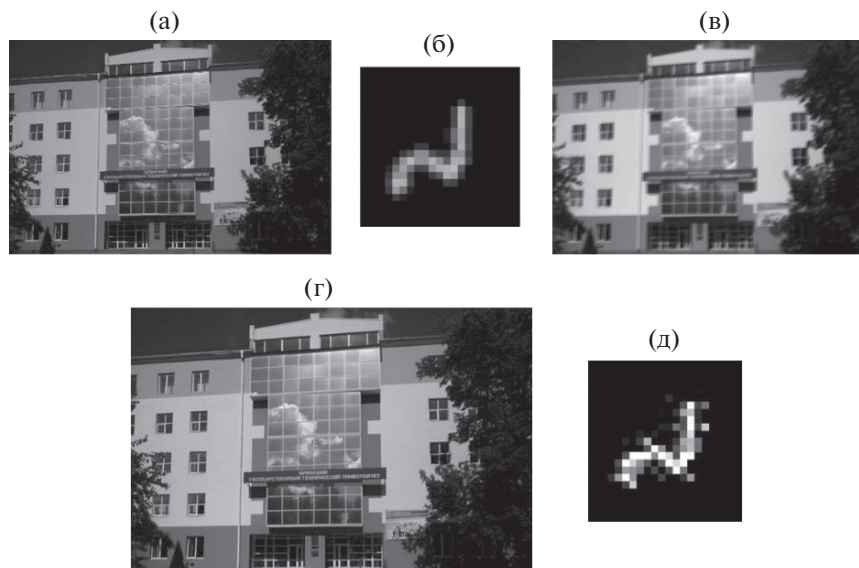


Рис. 5. Блок-схема алгоритма.

точно хорошие результаты по сравнению с другими классическими методами. В 90% случаев результат был не хуже, а в ряде случаев гораздо лучше, чем у классических методов (с меньшим количеством артефактов).

Дальнейшим развитием предложенного метода можно считать дополнительные исследования в области метрики оценки качества восстановленного результата, что может помочь достичь еще более качественных результатов, особенно в случаях искажения сложной формы.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Chen L., Fang F., Wang T., Zhang G.* Blind Image Deblurring with Local Maximum Gradient Prior // 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2019. P. 1742–1750.
2. *Bai Y., Cheung G., Liu X., Gao W.* Graph-Based Blind Image Deblurring From a Single Photograph // ArXiv, 2018.
3. *Levin A., Weiss Y., Durand F., Freeman W.T.* Understanding and evaluating blind deconvolution algorithms // 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009. P. 1964–1971.
4. *Gong Dong, Zhang Zhen, Shi Qinfeng, Anton van den Hengel, Shen Chunhua, Zhang Yanning.* Learning Deep Gradient Descent Optimization for Image Deconvolution // ArXiv, 2018.
5. *Cho S., Lee S.* Convergence Analysis of MAP Based Blur Kernel Estimation // 2017 IEEE International Conference on Computer Vision (ICCV). 2017. P. 4808–4816.
6. *Bai Y., Jia H., Jiang M. et al.* Single Image Blind Deblurring Using Multi-Scale Latent Structure Prior // ArXiv, 2019.
7. *Roy Prasun, Ghosh Subhankar, Bhattacharya Saumik, Pal Umapada.* Effects of Degradations on Deep Neural Network Architectures // ArXiv, 2019.
8. *Pan J., Hu Z., Su Z., Yang M.* Deblurring Text Images via L0-Regularized Intensity and Gradient Prior // 2014 IEEE Conference on Computer Vision and Pattern Recognition. 2014. P. 2901–2908.
9. *Вольнский М.А., Гуров И.П., Ермолаев П.А.* Методы компьютерной фотоники. СПб.: Университет ИТМО, 2014. 54 с.
10. *Гонсалес Р., Вудс Р.* Цифровая обработка изображений. Издание 3-е, исправленное и дополненное. М.: Техносфера, 2012. 1104 с.
11. *Giannakis G.B., Heath R.W.* Blind identification of multichannel FIR blurs and perfect image restoration // IEEE Transactions on Image Processing. 2000. V. 9. № 11. P. 1877–1896.
12. *Hosseini M.S., Plataniotis K.N.* Convolutional Deblurring for Natural Imaging // IEEE Transactions on Image Processing. 2020. V. 29. P. 250–264.
13. *Fergus R., Singh Barun, Hertzmann A., Roweis S.T., Freeman W.T.* Removing camera shake from a single image // SIGGRAPH, 2006.
14. *Chan T.F., Chiu-Kwong Wong.* Total variation blind deconvolution // IEEE Transactions on Image Processing. 1998. V. 7. № 3. P. 370–375.
15. *Lagendijk R.L., Tekalp A.M., Biemond J.* Maximum-Likelihood Image and Blur Identification – A Unifying Approach // Optical Engineering. 1990. V. 29. № 5. P. 422–435.
16. *Kim T., Lee K.M.* Segmentation-Free Dynamic Scene Deblurring // 2014 IEEE Conference on Computer Vision and Pattern Recognition. 2014. P. 2766–2773.
17. *Faramarzi E., Rajan D., Christensen M.P.* Unified Blind Method for Multi-Image Super-Resolution and Single/Multi-Image Blur Deconvolution // IEEE Trans-

- actions on Image Processing. 2013. V. 22. № 6. P. 2101–2114.
18. *Xiao L., Gregson J., Heide F., Heidrich W.* Stochastic Blind Motion Deblurring // 2015 IEEE Transactions on Image Processing. 2015. V. 24. № 10. P. 3071–3085.
  19. *Gregson J., Heide F., Hullin M.B. et al.* Stochastic deconvolution. // 2013 IEEE Conference on Computer Vision and Pattern Recognition. 2013. P. 1043–1050.
  20. *Babacan S.D., Molina R., Do M.N., Katsaggelos A.K.* Bayesian Blind deconvolution with general sparse image priors // Springer European Conference on Computer Vision. 2012. P. 341–355.
  21. *Tzikas D.G., Likas A.C., Galatsanos N.P.* Variational Bayesian Sparse Kernel-Based Blind Image Deconvolution with Student's-t Priors // 2009 IEEE Transactions on Image Processing. 2009. V. 18. № 4. P. 753–764.
  22. *Zuo W., Ren D., Zhang D. et al.* Learning Iteration-wise Generalized Shrinkage Thresholding Operators for Blind Deconvolution // 2016 IEEE Transactions on Image Processing. 2016. V. 25. № 4. P. 1751–1764.
  23. *Perrone D., Favaro P.* A Clearer Picture of Total Variation Blind Deconvolution // 2016 IEEE Transactions on Pattern Analysis and Machine Intelligence. 2016. V. 38. № 6. P. 1041–1055.
  24. *Simoncelli E.P.* Bayesian Denoising of Visual Images in the Wavelet Domain // Lecture Notes in Statistics. 1999. V. 141. P. 291–308.
  25. *Weiss Y., Freeman W.T.* What makes a good model of natural images? // 2007 IEEE Conference on Computer Vision and Pattern Recognition. 2007. P. 1–8.
  26. *Roth S., Black M.J.* Fields of Experts: a framework for learning image priors // 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). 2005. V. 2. P. 860–867.
  27. *Perrone D., Favaro P.* Total Variation Blind Deconvolution: The Devil Is in the Details // 2014 IEEE Conference on Computer Vision and Pattern Recognition. 2014. P. 2909–2916.
  28. *Padcharoen A., Kitkuan D., Kumam P. et al.* Accelerated alternating minimization algorithm for Poisson noisy image recovery // Inverse Problems in Science and Engineering. 2020. P. 1–26.
  29. *Carbajal G., Patricia V., Mauricio D. et al.* Non-uniform Blur Kernel Estimation via Adaptive Basis Decomposition // ArXiv. 2021.
  30. *Tran P., Tran A., Phung Q., Hoai M.* Explore Image Deblurring via Blur Kernel Space // ArXiv. 2021.
  31. *Hu X., Ren W., Yu K.* Pyramid Architecture Search for Real-Time Image Deblurring // Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2021. P. 4298–4307.
  32. *Kupyn O., Budzan V., Mykhailych M. et al.* DeblurGAN: Blind Motion Deblurring Using Conditional Adversarial Networks // ArXiv. 2017.
  33. *Wen Y., Chen J., Sheng B. et al.* Structure-Aware Motion Deblurring Using Multi-Adversarial Optimized CycleGAN // 2021 IEEE Transactions on Image Processing. 2021. V. 30. P. 6142–6155.
  34. *Zhou Wang, Bovik A.C., Sheikh H.R., Simoncelli E.P.* Image quality assessment: from error visibility to structural similarity // IEEE Trans. on Image Processing. 2004. V. 13. № 4. P. 600–612. <https://doi.org/10.1109/TIP.2003.819861>
  35. *Eyal W.* Hands-On Genetic Algorithms with Python. Birmingham, UK: Packt Publishing Ltd., 2020.