

Для решения ряда задач обработки (например восстановления) изображений необходимо знание оператора системы. Этот оператор рассчитывают на этапе проектирования с использованием физических и математических моделей видеотракта. Вследствие неизбежных упрощений, связанных как с недостаточной изученностью явлений, так и с вычислительными аспектами оператор системы нуждается в уточнении. Для этого проводят испытания, заключающиеся в «предъявлении» на вход системы известного изображения. Затем по входному изображению и результатам его регистрации на выходе системы строится оптимальная, в некотором смысле, модель. Эта задача известна как идентификация систем. Обычно она формулируется (в узком смысле [30]) как задача оценки параметров модели системы формирования изображений из априори заданного класса. Наиболее широко используются описанные линейные модели типа

$$g(n_1, n_2) = \sum_{m_1=-\infty}^{\infty} \sum_{m_2=-\infty}^{\infty} h(m_1, m_2) f(n_1 - m_1, n_2 - m_2) \quad (1)$$

и/или

$$g(n_1, n_2) = \sum_{(m_1, m_2) \in Q_g} a_{m_1, m_2} g(n_1 - m_1, n_2 - m_2) + \sum_{(m_1, m_2) \in Q_f} b_{m_1, m_2} f(n_1 - m_1, n_2 - m_2) \quad (2)$$

с постоянными по пространственным координатам параметрами (ЛПП-системы). Вычислительные преимущества, связанные с применением таких моделей, очевидны. Если в действительности искажения оказываются пространственно-зависимыми (неизопланатичными), то, как указывалось выше, линейные модели с постоянными параметрами строят на малых фрагментах изображений.

Для формулировки задачи идентификации указанных линейных моделей мы должны внести в уравнения (1), (2) некоторые важные дополнения. Во-первых, нельзя оценить бесконечное число значений импульсной характеристики, то есть число слагаемых в правой части (1) всегда ограничено. Во-вторых, задача идентификации решается по результатам измерений, которые всегда содержат погрешности. С учетом сказанного уравнения перепишем, соответственно, в виде

$$g(n_1, n_2) = \sum_{(m_1, m_2) \in Q_f} h_{m_1, m_2} f(n_1 - m_1, n_2 - m_2) + \delta g(n_1, n_2), \quad (3)$$

$$\begin{aligned} g(n_1, n_2) = & \sum_{(m_1, m_2) \in Q_g} a_{m_1, m_2} g(n_1 - m_1, n_2 - m_2) + \\ & + \sum_{(m_1, m_2) \in Q_f} b_{m_1, m_2} f(n_1 - m_1, n_2 - m_2) + \xi(n_1, n_2). \end{aligned} \quad (4)$$

В (3), (4) используются те же обозначения, что и в (1), (2). Дополнительно введены лишь обозначения для ошибок  $\delta g(n_1, n_2)$  (измерений, ограничений на порядок модели и др.), а фигурирующая в (4) ошибка  $\xi(n_1, n_2)$  определяется как

$$\xi(n_1, n_2) = \delta g(n_1, n_2) - \sum_{(m_1, m_2) \in Q_g} a_{m_1, m_2} \delta g(n_1 - m_1, n_2 - m_2). \quad (5)$$

Модель, описываемую уравнением (3), в соответствии с принятой терминологией далее будем называть КИХ-фильтром, а модель, соответствующую уравнению (4) — БИХ-фильтром. Поскольку задача идентификации должна решаться по совокупности отсчетов  $f(n_1, n_2)$  входного (неискаженного) и  $g(n_1, n_2)$  выходного (искаженного) изображений,

для дальнейшего изложения удобно от уравнений (3), (4) перейти к их матричным представлениям.

Рассмотрим уравнение (4). Предположим, что общее число коэффициентов  $\{a_{m_1, m_2}\}$ ,  $\{b_{m_1, m_2}\}$  этого уравнения равно  $M$ , а фрагмент на выходном изображении содержит  $N$  различных отсчетов  $g(n_1, n_2)$ . Отсчеты, фигурирующие в правой части (4) при одном из фиксированных положений  $(n_1, n_2)$  выходного отсчета, можно представить в виде элементов вектор-строки:

$$\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,M}] = [\dots, g(n_1 - m_1, n_2 - m_2), \dots, f(n_1 - m_1, n_2 - m_2), \dots], \quad (6)$$

$$(m_1, m_2) \in Q_g \cup Q_f, i = \overline{1, N},$$

а соответствующий вектор искомых параметров будет иметь вид

$$\mathbf{c} = [c_1, c_2, \dots, c_M]^T = [\dots, a_{m_1, m_2}, \dots, b_{m_1, m_2}, \dots]^T, (m_1, m_2) \in Q_g \cup Q_f, i = \overline{1, N}. \quad (7)$$

Заметим, что размерность вектора искомых параметров зависит от порядка передаточной функции системы, то есть от размеров опорных областей как на входном, так и на выходном изображении. Для  $N$  различных положений опорных областей из векторов-строк (6) составим  $N \times M$ -матрицу  $\mathbf{X}$ , а из  $N$  отсчетов  $g(n_1, n_2)$  на выходном изображении и  $N$  соответствующих им ошибок  $\xi(n_1, n_2)$  составим  $N \times 1$ -векторы  $\mathbf{y}$  и  $\boldsymbol{\xi}$  соответственно. Если вектор параметров  $\mathbf{c}$  остается неизменным при любом положении опорной области на фрагменте (для ЛПП-систем это всегда выполняется), с использованием введенных обозначений можно записать следующее матричное равенство:

$$\mathbf{y} = \mathbf{X}\mathbf{c} + \xi. \quad (8)$$

Задача заключается в том, чтобы по одной реализации (фрагменту изображения) построить оценку  $\hat{\mathbf{c}}$  вектора параметров  $\mathbf{c}$  по доступным для непосредственного наблюдения  $N \times M$ -матрице  $\mathbf{X}$  и  $N \times 1$ -вектору  $\mathbf{y}$  ( $N > M$ ), при неизвестном  $N \times 1$ -векторе ошибок  $\xi$ .

Аналогичное матричное равенство можно построить для модели (3) КИХ-фильтра. Сопоставив приведенные выше обозначения с (3) нетрудно заметить, что вектор искомых параметров  $\mathbf{c}$  в данном случае представляется в виде

$$\mathbf{c} = [c_1, c_2, \dots, c_M]^T = [\dots, h_{m_1, m_2}, \dots]^T, (m_1, m_2) \in Q_f$$

а каждая строка матрицы  $\mathbf{X}$  состоит из отсчетов только входного изображения:

$$\mathbf{x}_i = [x_{i,1}, x_{i,2}, \dots, x_{i,M}] = [\dots, f(n_1 - m_1, n_2 - m_2), \dots], (m_1, m_2) \in Q_f, i = \overline{1, N}.$$

Постановка задачи идентификации модели КИХ-фильтра формально совпадает с задачей идентификации БИХ-фильтра. Важные отличия состоят в следующем. Компоненты  $N \times 1$ -вектора ошибок  $\xi$  в данном случае не зависят от полезных сигналов (отсчетов поля яркости выходного изображения), как это имеет место в (5) для БИХ-фильтра. Кроме того, размерность вектора искомых параметров зависит лишь от размеров опорной области на входном изображении. Это оказывается существенным для рассматриваемых методов. Подчеркнем, что с точки зрения задачи идентификации порядок обхода точек на фрагменте (изображении) не играет роли. Это приводит лишь к перестановке строк в уравнении (8). В то же время применение оцененных моделей КИХ и БИХ-фильтров существенно различается.

В данной дипломной работе рассматривается нелинейная фильтрация изображения с линейной по параметрам моделью. Для изучения была выбрана маска размером  $5 \times 5$ :

6	5	4	5	6
5	3	2	3	5
4	2	1	2	4
5	3	2	3	5
6	5	4	5	6

Таблица 1. Маска.

и функция вида:

$$f(x) = x + x^2 + x^3 \quad (1)$$

Тогда для данной маски, уравнение выходного сигнала будет иметь следующий вид:

$$y = a_1x_1 + a_2x_2 + a_3x_2^2 + a_4x_2^3 + a_5x_3 + a_6x_3^2 + a_7x_3^3 + a_8x_4 + \\ + a_9x_4^2 + a_{10}x_4^3 + a_{11}x_5 + a_{12}x_5^2 + a_{13}x_5^3 + a_{14}x_6 + a_{15}x_6^2 + a_{16}x_6^3, \quad (2)$$

где  $a_i, i = \overline{1,16}$  — идентифицируемые параметры,  $x_i$  — среднее значение яркости пикселей изображения с индексом соответствующим значению в маске (Таблица 1), т.е. равноудаленные от центра маски значения.

Таким образом первые три слагаемых выглядят следующим образом:

$$y = a_1x_{i,j} + a_2 \frac{(x_{i-1,j-1} + x_{i-1,j+1} + x_{i+1,j+1} + x_{i+1,j-1})}{4} + \\ + a_3 \frac{(x_{i-1,j-1} + x_{i-1,j+1} + x_{i+1,j+1} + x_{i+1,j-1})^2}{16} \quad (3)$$

Идентификация параметров модели происходит на основе исходного изображения и размытого. На исходном изображении случайным образом выбирается  $N$  точек, и выбирается набор соответствующих точек на размытом изображении. Далее строится система из  $N$  линейных уравнений, как описано выше, где матрица  $\mathbf{X}$  состоит из значений  $x_i$  из уравнения (2) для маски с центром в случайно выбранной точке, а вектор  $\mathbf{y}$  состоит из значений яркости, случайно выбранных точек, исходного изображения. Производится проверка на наличие решения данной системы, согласно теореме Кронекера-Капелли: система линейных алгебраических уравнений совместна тогда и только тогда, когда ранг её основной матрицы равен рангу её расширенной матрицы, причём система имеет единственное решение, если ранг равен числу неизвестных и бесконечное множество решений, если ранг меньше числа неизвестных, т.е. для нашей системы должно выполняться следующее равенство:

$$\text{rang } \mathbf{X} = 16. \quad (4)$$

Затем по простой формуле находится оценка вектора решений  $\hat{a}$ :

$$\hat{a} = [\mathbf{X}^T \mathbf{X}]^{-1} \mathbf{X}^T \mathbf{y}. \quad (5)$$

Также осуществляется проверка критерия того, чтобы сумма значений вектора параметров было около 1; чтобы не сильно менялась яркость обрабатываемой области.

По найденным значениям параметров, и по формуле (1) можно восстанавливать изображения, применяя маску (Таблица 1), ко всем пикселям изображения.

## Практические результаты

Для реализации алгоритма нелинейной фильтрации с линейной по параметрам моделью была реализована программа в математическом пакете Matlab. Сначала считывается картинка (исходное изображение), затем из нее выбирается один канал цвета, чтобы дальше работать с черно-белой картинкой. Далее для исходное изображение подвергается искажению – размытие. Во время выполнения дипломной работы было реализовано несколько алгоритмов размытия: размытие с помощью КИХ-фильтра, размытие с помощью БИХ-фильтра, фильтр низких частот Гаусса. Визуально было определено, что лучший визуальный эффект размытия дает фильтр низких частот Гаусса. Данный фильтр создается и применяется с помощью стандартных функции пакета Matlab: *fspecial('gaussian', hsize, sigma)* и *imfilter(image, filter, 'replicate')*. После этого запускается алгоритм идентификации параметров модели по искаженному и исходному изображению (описанный выше). После чего происходит попытка восстановления искаженного изображения, для оценки качества работы фильтра.

Реализована также программа, которая по найденным параметрам модели восстанавливает произвольно поданную картинку на вход. Принцип работы следующий: происходит считывание картинки (исходное изображение), затем из нее выбирается один канал цвета, чтобы дальше работать с черно-белой картинкой. Далее она подвергается искажению, аналогичному, что и при выполнении первой части: поиска параметров модели. Затем для известных параметров происходит попытка восстановления искаженной картинки. На протяжении работы программы выводятся

сообщения об окончании каждого этапа обработки картинки, и каждая картинка выводится в отдельном окне, для возможности сравнения результата.

Также был реализован простейший линейный фильтр для возможности сравнения нелинейного фильтра с линейным. Линейный фильтр для той же маски, что и в случае нелинейного имеет вид:

$$y = a_1x_1 + a_2x_2 + \dots + a_{25}x_{25} \quad (1)$$

где  $a_i, i = \overline{1, 25}$  - неизвестные параметры модели,  $x_i, i = \overline{1, 25}$  - значение яркости каждого пикселя маски. Для данного фильтра реализован аналогичный алгоритм нахождения вектора оценок параметров модели по исходному и искаженному изображениям.

Рассмотри результат работы программы для различной силы размытия изображения (различное значение  $\sigma$  при применении фильтра низких частот Гаусса).

Ниже проиллюстрированы результаты для слабого размытия (значение  $\sigma = 1.2$ ):

Вектор параметров нелинейного фильтра:

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$a_6$	$a_7$	$a_8$
5.1902	-3.8189	0.0808	-0.0002	0.3009	-0.1165	0.0003	0.7376
$a_9$	$a_{10}$	$a_{11}$	$a_{12}$	$a_{13}$	$a_{14}$	$a_{15}$	$a_{16}$
-0.0414	0.0001	-2.7749	0.0988	-0.0003	1.357	-0.0216	7.5947e-05

Таблица 2. Значения вектора параметров нелинейного фильтра при слабом размытии.

Сумма значений равна 0.9919.

Вектор параметров линейного фильтра:



$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
6.3205	0.3811	2.1263	-0.2268	-0.4478
$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$
-2.1117	-3.019	-1.9179	-2.8602	-0.0817
$a_{11}$	$a_{12}$	$a_{13}$	$a_{14}$	$a_{15}$
-1.184	0.11526	0.5387	-0.2225	0.673
$a_{16}$	$a_{17}$	$a_{18}$	$a_{19}$	$a_{20}$
0.105	0.0674	0.2323	0.1724	0.5715
$a_{21}$	$a_{22}$	$a_{23}$	$a_{24}$	$a_{25}$
-1.0903	-0.1789	0.5733	0.5594	-0.1548

*Таблица 3. Значения вектора параметров линейного фильтра при слабом размытии.*

Сумма значений равна 1.0026.



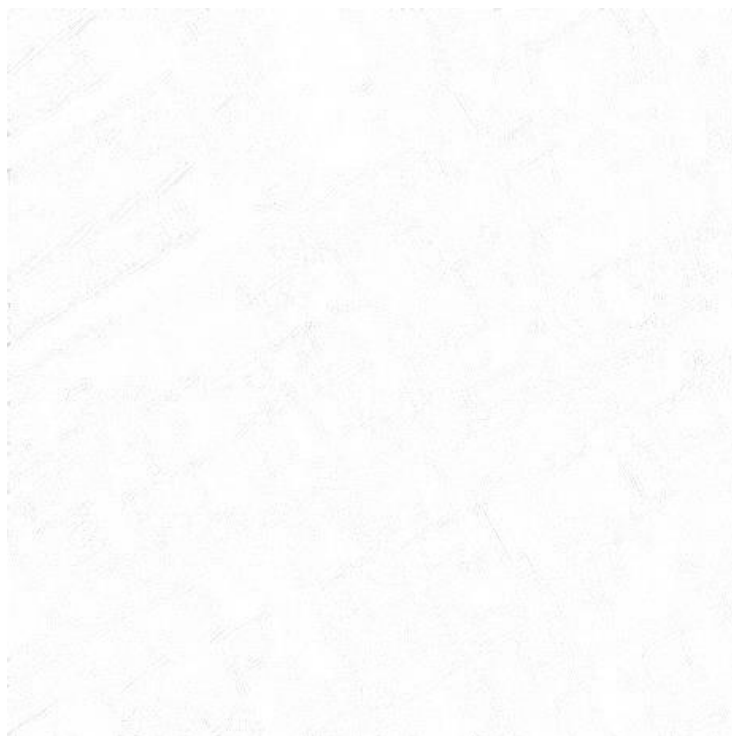
*Изображение 1. Исходное изображение.*



*Изображение 2. Размытое изображение.*



*Изображение 3. Восстановленное с помощью нелинейного фильтра изображение.*

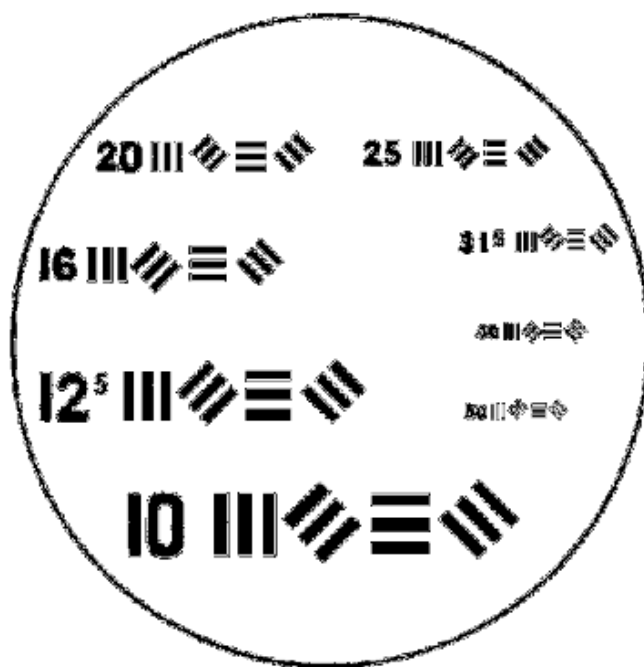


*Изображение 4. Карта разности между исходным и восстановленным изображениями.*



*Изображение 5. Восстановленное с помощью линейного фильтра изображение.*

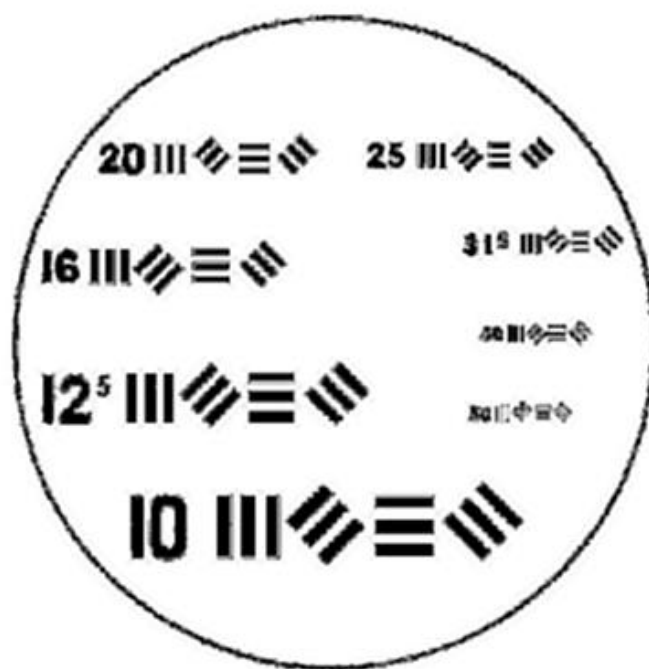
Можно видеть, что нелинейный фильтр дает неплохой результат. Согласно карте разности между исходным и восстановленным изображениями мало заметная разница на краях объектов изображения. Но так же можно заметить, что очень мала разница между изображением восстановленным нелинейным фильтром и изображением восстановленным линейным фильтром. Далее проиллюстрированы результаты восстановления другого изображения, но для уже найденных параметров модели, т.е. задача идентификации параметров не решалась.



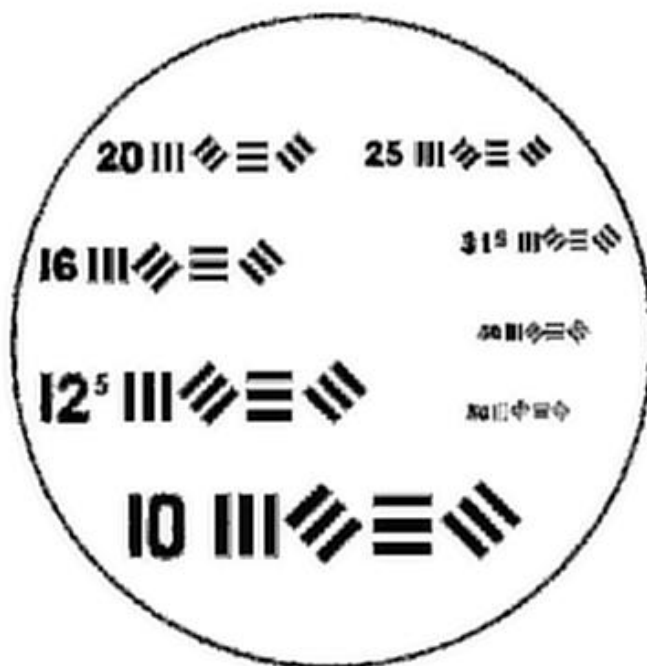
*Изображение 6. Исходное изображение.*



Изображение 7. Размытое изображение.



Изображение 8. Восстановленное с помощью нелинейного фильтра изображение.



*Изображение 9. Восстановленное с помощью линейного фильтра изображение.*

По представленным результатам хорошо видно, что для восстановления слабо размытых изображений фильтр работает очень хорошо, но незаметно разницы между линейным и нелинейным фильтрами.

Рассмотрим теперь результат для среднего размытия для таких же исходных изображений (значение параметра фильтра  $\sigma = 1.6$ ):

Вектор параметров нелинейного фильтра:

$a_1$	$a_2$	$a_3$	$a_4$
7.2442	-4.497	0.0739	-9.0658e-05
$a_5$	$a_6$	$a_7$	$a_8$
-2.4486	-0.05117	-1.8416e-05	0.6546
$a_9$	$a_{10}$	$a_{11}$	$a_{12}$
-0.0117	3.3038e-05	-0.395	-0.0546
$a_{13}$	$a_{14}$	$a_{15}$	$a_{16}$
0.0002	0.4408	0.0435	-0.0001

*Таблица 4. Значения вектора параметров нелинейного фильтра при слабом размытии.*

Сумма значений равна 0.9989.

Вектор параметров линейного фильтра:

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
6.4343	0.2448	0.9279	0.8472	0.0715
$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$
-1.1118	-2.2484	-1.8928	-1.605	1.9866
$a_{11}$	$a_{12}$	$a_{13}$	$a_{14}$	$a_{15}$
-0.3214	-0.1973	0.8368	-1.0219	0.1008
$a_{16}$	$a_{17}$	$a_{18}$	$a_{19}$	$a_{20}$
-1.0703	-1.0159	-2.5088	-1.2482	-0.4012
$a_{21}$	$a_{22}$	$a_{23}$	$a_{24}$	$a_{25}$
-0.0491	1.206	1.008	1.1151	0.9158

*Таблица 5. Значения вектора параметров линейного фильтра при слабом размытии.*

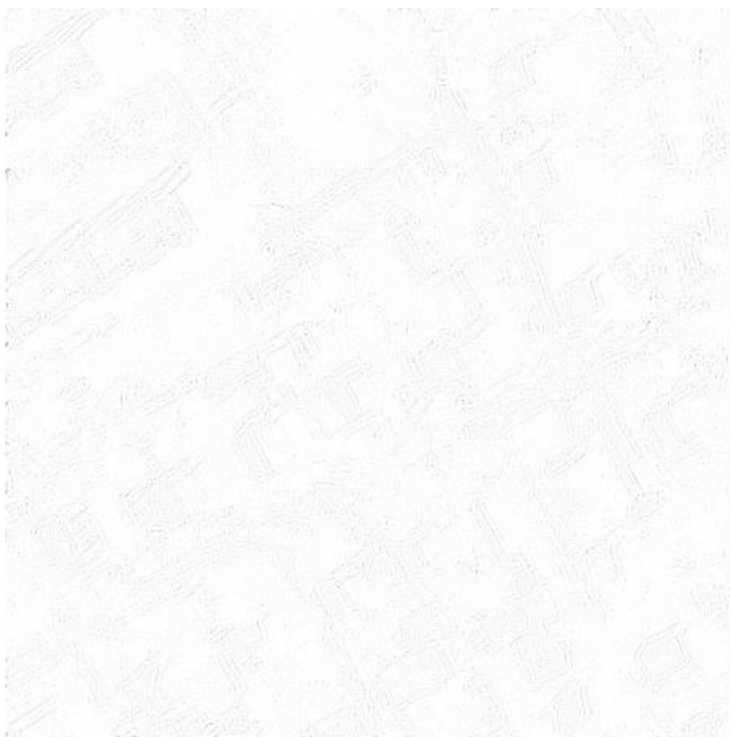
Сумма значений равна 1.0028.



*Изображение 10. Размытое изображение.*



*Изображение 11. Восстановленное с помощью нелинейного фильтра изображение.*



*Изображение 12. Карта разности между исходным и восстановленным изображениями.*





*Изображение 13. Восстановленное с помощью линейного фильтра изображение.*

Визуально результат аналогичен случаю слабого размытия. Рассмотрим теперь результаты для работы фильтров с известными параметрами для другого изображения.



Изображение 14. Размытое изображение.



Изображение 15. Восстановленное с помощью нелинейного фильтра изображение.



*Изображение 16. Восстановленное с помощью линейного фильтра изображение.*

Для восстановления средне размытых изображений фильтр работает также хорошо, но разница между линейным и нелинейными фильтрами практически незаметна. Лишь при пристальном рассмотрении увеличенного изображения можно увидеть, что нелинейный фильтр дает чуть более высокую четкость на краях объектов изображения.

Рассмотрим теперь результат для среднего размытия для таких же исходных изображений (значение параметра фильтра  $\sigma = 2$ ):

Вектор параметров нелинейного фильтра:

$a_1$	$a_2$	$a_3$	$a_4$
7.6485	-5.8515	0.1367	-0.0004
$a_5$	$a_6$	$a_7$	$a_8$
-0.1478	-0.0907	0.0003	5.1769
$a_9$	$a_{10}$	$a_{11}$	$a_{12}$
-0.0619	6.9113e-05	-7.2183	-0.0544
$a_{13}$	$a_{14}$	$a_{15}$	$a_{16}$
0.0002	1.4015	0.07	-0.0002

*Таблица 6. Значения вектора параметров нелинейного фильтра при слабом размытии.*

Сумма значений равна 1.0089.

Вектор параметров линейного фильтра:

$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
7.7786	2.9204	1.8288	3.081	2.5833
$a_6$	$a_7$	$a_8$	$a_9$	$a_{10}$
-3.0844	-3.208	-2.0024	-2.9293	-3.2013
$a_{11}$	$a_{12}$	$a_{13}$	$a_{14}$	$a_{15}$
0.1598	-0.01	-2.142	-2.5103	-0.19
$a_{16}$	$a_{17}$	$a_{18}$	$a_{19}$	$a_{20}$
0.2858	-3.6973	-0.0708	-0.6806	-1.5225
$a_{21}$	$a_{22}$	$a_{23}$	$a_{24}$	$a_{25}$
-2.4002	1.3616	1.8153	3.0038	3.8354

*Таблица 7. Значения вектора параметров линейного фильтра при слабом размытии.*

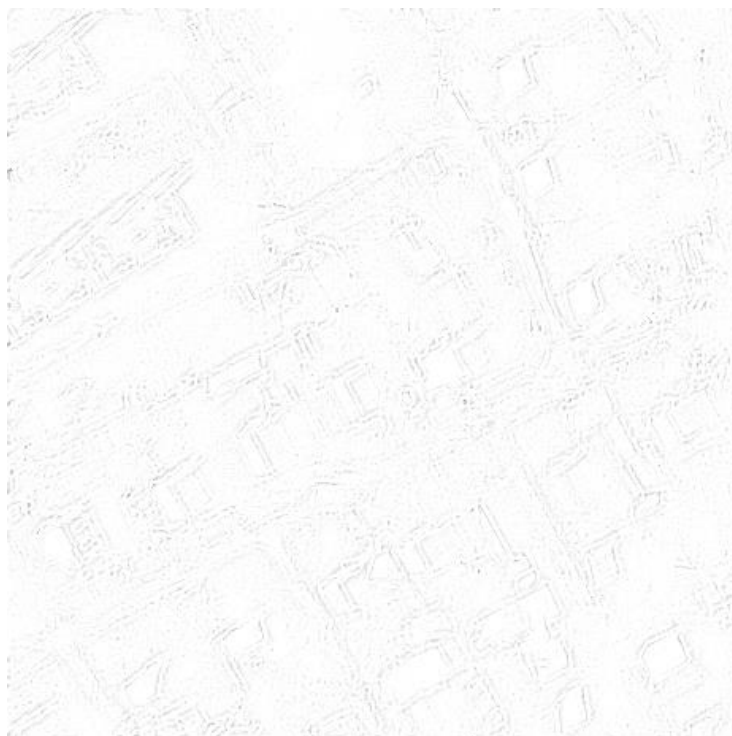
Сумма значений равна 1.0046.



*Изображение 17. Размытое изображение.*



*Изображение 18. Восстановленное с помощью нелинейного фильтра изображение.*



*Изображение 19. Карта разности между исходным и восстановленным изображениями.*



*Изображение 20. Восстановленное с помощью линейного фильтра изображение.*

Из результатов видно, что на карте разности отмечено намного больше ярких точек, чем ранее. А также более четко видно разницу между линейным и нелинейным фильтром: результат работы нелинейного фильтра выдал более четкую картинку, чем линейный фильтр. В результате чего можно судить о преимуществе нелинейного фильтра. Но прежде рассмотрим результаты для найденных параметров модели для другого изображения:



*Изображение 21. Размытое изображение.*



Изображение 22. Восстановленное с помощью нелинейного фильтра изображение.



Изображение 23. Восстановленное с помощью линейного фильтра изображение.



По представленным результатам можно судить о преимуществе использования нелинейного фильтра перед линейным.

И так практически было показано достоинство использования нелинейной фильтрации с линейной по параметрам моделью.