

РЕАЛИЗАЦИЯ SD-ROM ФИЛЬТРА НА ОСНОВЕ КОНЦЕПЦИИ НЕЧЕТКОЙ ЛОГИКИ

Рассмотрена реализация SD-ROM фильтра с использованием нечеткой логики. В результате проведенного сравнительного анализа SD-ROM и медианных фильтров в задачах удаления импульсного шума подтверждается превосходство SD-ROM фильтрации. Применение концепции нечеткой логики при реализации SD-ROM фильтров позволяет практически полностью удалить шум на изображении.

Введение

Медианный фильтр является наиболее используемым нелинейным фильтром для удаления импульсного шума. Тем не менее при его применении изображение подвергается сглаживанию, а детали изображения становятся расплывчатыми, особенно при использовании окна фильтра размером более 3×3 . Для улучшения характеристик медианной фильтрации разработано много различных модификаций, включая центрированные и перестановочные взвешенные медианные фильтры [1]. Хотя данные модификации и превосходят обычную медианную фильтрацию по своим характеристикам, при большом уровне шума, вероятность которого превышает 0,3, данные фильтры не способны эффективно справиться с задачей его удаления с сохранением границ деталей изображения.

С целью решения поставленной задачи разработан SD-ROM фильтр [2, 3], который также называют зависимым от сигнала фильтром среднего значения ранжированной последовательности.

SD-ROM фильтр

Рассмотрим окно изображения, размером 3×3 , в котором центральный элемент обозначим $x(5)$. Определим вектор $x'(n)$, который состоит из элементов $x(n)$ рассматриваемого окна, за исключением центрального элемента $x(5)$:

$$\begin{aligned} x'(n) &= [x'_1(n), x'_2(n), \dots, x'_8(n)] = \\ &= [x(n_1 - 1, n_2 - 1), x(n_1 - 1, n_2), x(n_1 - 1, n_2 + 1); \\ &\quad x(n_1, n_2 - 1), x(n_1, n_2 + 1); \\ &\quad x(n_1 + 1, n_2 - 1), x(n_1 + 1, n_2), x(n_1 + 1, n_2 + 1)], \end{aligned} \quad (1)$$

где n_1 представляет собой координаты по оси y , n_2 – координаты по оси x , если рассматривать изображение в декартовых координатах.

Полученные элементы вектора $x'(n)$ можно отсортировать по величине в порядке возрастания, т.е. по рангу, при этом получим вектор ранжированной последовательности:

$$r(n) = [r_1(n), r_2(n), \dots, r_8(n)], \quad (2)$$

где $r_1(n) \leq r_2(n) \leq \dots \leq r_8(n)$.

После этого обозначим $m(n)$ как среднее значение ранжированной последовательности (ROM – Rank Ordered Mean):

$$m(n) = \frac{r_4(n) + r_5(n)}{2}. \quad (3)$$

Следует отметить, что (3) представляет собой выход простого медианного фильтра с той лишь разницей, что на его вход поступают все элементы окна за исключением центрального элемента.

Далее определим вектор $d(n) \in R^4$, который назовем вектором разниц:

$$d(n) = [d_1(n), d_2(n), d_3(n), d_4(n)], \quad (4)$$

где

$$d_i(n) = \begin{cases} r_i(n) - x(5), & \text{для } x(5) \leq m(n), \\ x(5) - r_i(n), & \text{для } x(5) > m(n), \end{cases} \quad (5)$$

при $i = 1, 2, 3, 4$.

Вектор разниц представляет собой оценку вероятности воздействия шума на элемент $x(5)$, находящийся в центре окна. Например, рассмотрим элемент $d_1(n)$: если он положителен, значит текущий элемент $x(5)$ является либо наибольшим, либо наименьшим в рассматриваемом окне. Если значение $d_1(n)$ не только положительное, но и превышает определенное пороговое значение, то вероятность того, что элемент $x(5)$ подвержен влиянию шума, очень высока.

Таким образом, с помощью четырех элементов вектора $d(n)$ можно более точно оценить вероятность воздействия импульсного шума на центральный элемент даже в том случае, если в окне фильтра присутствует не один элемент, подверженный влиянию шума.

Для изображений, подверженных влиянию униполярного импульсного шума, величина которого определяется максимальным значением яркости, $d_i(n)$ принимает вид

$$d_i(n) = x(5) - r_i(n), \quad i = 1, 2, 3, 4. \quad (6)$$

Схожим образом при униполярном шуме, величина которого определяется минимальным значением яркости, $d_i(n)$ принимает вид

$$d_i(n) = r_i(n) - x(5), \quad i = 1, 2, 3, 4. \quad (7)$$

На рисунке 1 изображена структура SD-ROM фильтра.

Оператор определения импульса в зависимости от входной последовательности принимает одно из двух решений: если центральный элемент окна подвержен влиянию шума, то к входной последовательности применяется ROM фильтр, в противном случае выходом фильтра является исходный центральный элемент.

Рассмотрим подробно алгоритм оператора определения импульса. Данный оператор определяет $x(5)$ как элемент, не подверженный влиянию шума, если выполняются следующие неравенства для всех i :

$$d_i(n) \leq T_i, \quad i = 1, 2, 3, 4, \quad (8)$$

где T_1, T_2, T_3, T_4 – пороговые коэффициенты, причем

$$T_1 < T_2 < T_3 < T_4. \quad (9)$$

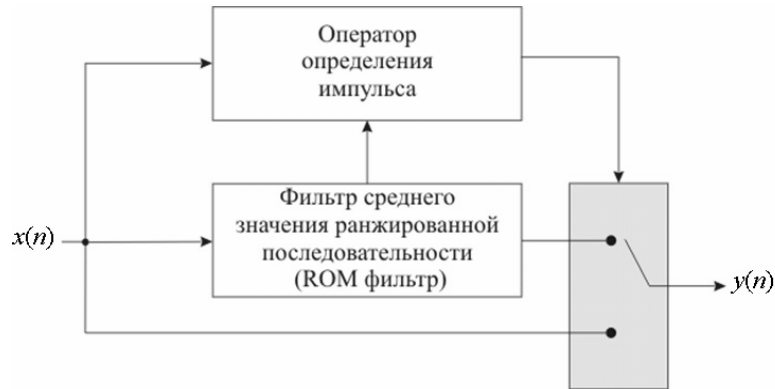


Рис. 1 SD-ROM фильтр

Если неравенство (8) не выполняется для всех i , рассматриваемый элемент считается подверженным влиянию шума.

Рассмотренный SD-ROM фильтр также называют пороговым SD-ROM фильтром, т.к. его выход определяется только вектором разниц (4) и пороговыми коэффициентами. Результаты использования данного алгоритма для удаления импульсного шума [4] позволили выделить для каждого порогового коэффициента T_i множество значений, при которых SD-ROM фильтр наилучшим образом удаляет импульсный шум:

$$\begin{aligned} T_1 &\in \{4, 8, 12\}, \\ T_2 &\in \{15, 20, 25\}, \\ T_3 &\in \{38, 40, 42\}, \\ T_4 &\in \{47, 50, 53\}. \end{aligned} \quad (10)$$

В большинстве случаев используют следующие пороговые коэффициенты:

$$T_1 = 8, \quad T_2 = 20, \quad T_3 = 40, \quad T_4 = 50. \quad (11)$$

Тем не менее в некоторых приложениях требуется рассматривать пороговые коэффициенты в следующих интервалах:

$$\begin{aligned} 5 &\leq T_1 \leq 15, \\ 15 &\leq T_2 \leq 25, \\ 30 &\leq T_3 \leq 50, \\ 40 &\leq T_4 \leq 60. \end{aligned} \quad (12)$$

Процесс выбора пороговых коэффициентов, хоть и является экспериментальным, но не требует большого количества экспериментов, т.к. наблюдатель уверенно различает уровень импульсного шума.

Следует отметить существование других схем для удаления импульсного шума [5–9]. Основной особенностью этих схем, так же как и при рассмотрении порогового SD-ROM фильтра, является тот факт, что выход фильтра принимает либо входное значение, либо вычисляется с использованием различных модификаций медианной фильтрации.

Рассмотренный SD-ROM фильтр находит применение не только в задачах обработки изображений, но и в других приложениях, включая обработку многомерных и одномерных сигналов [10].

Так же как и в случае обычной медианной фильтрации, существуют рекурсивные SD-ROM фильтры, использование которых позволяет достичь лучших результатов в сравнении с нерекурсивными. Реализация рекурсивного алгоритма подобна реализации рекурсивной взвешенной медианной фильтрации, т.е. на вход фильтра подаются как элементы исходного изображения, так и элементы предыдущего выхода фильтра.

Использование нечеткой логики в SD-ROM фильтрах

Как видно из рисунка 1, выход фильтра $y(n)$ меняется в зависимости от переключателя (обозначен серым прямоугольником), которому подает команду на переключение оператор определения импульса. Основными данными, на основе которых оператор переключает схему, являются вектор разниц (4), пороговые коэффициенты, выход фильтра среднего значения ранжированной последовательности и вектор входных значений. Используя концепцию нечеткой логики [11], данный метод можно записать в общем виде, где $y(n)$ представляет собой линейную комбинацию $x(n)$ и $m(n)$:

$$y(n) = \alpha(d(n))x(n) + \beta(d(n))m(n), \quad (13)$$

где сумма весовых коэффициентов $\alpha(d(n))$ и $\beta(d(n))$ равна единице, т.е.

$$\alpha(d(n)) = 1 - \beta(d(n)). \quad (14)$$

Заметим, что коэффициенты $\alpha(d(n))$ и $\beta(d(n))$ имеют отношение только к вектору разниц $d(n)$, а не к пороговым коэффициентам.

Данная реализация также известна как «обобщенный SD-ROM фильтр» и схематично изображена на рисунке 2.

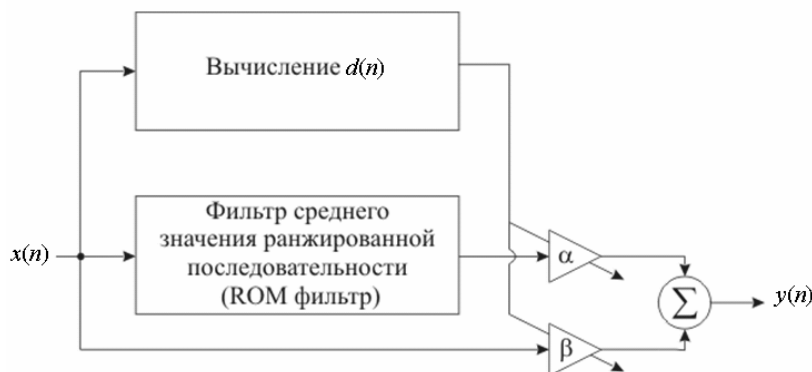


Рис. 2 Обобщенный SD-ROM фильтр

Стоит отметить, что пороговый SD-ROM фильтр является частным случаем обобщенного SD-ROM фильтра, где $\alpha(d(n))$ может принимать только два значения: 0 и 1. Различие между двумя фильтрами можно наблюдать при сравнении функций $\alpha(d(n))$ (рис. 3, 4), где для наглядности рассматривается двухмерный вектор разниц, т.е. $d(n) = [d_1(n), d_2(n)]$.

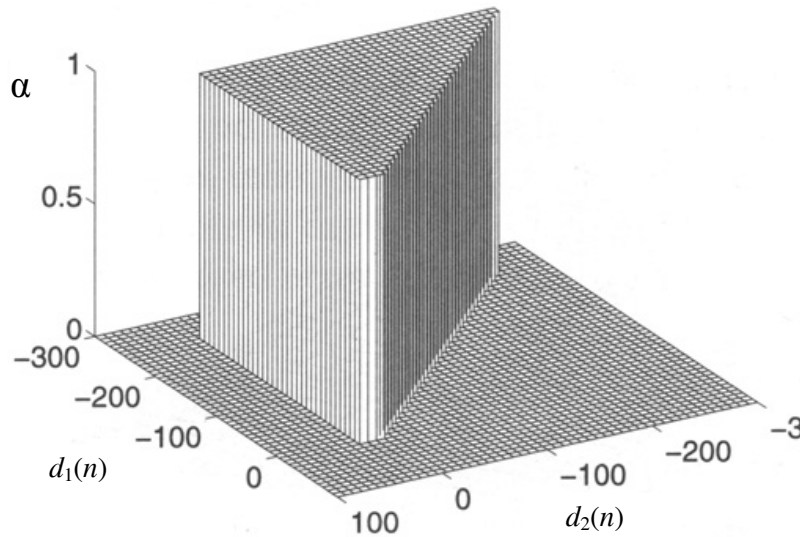


Рис. 3 Функция $\alpha(d(n))$ для порогового SD-ROM фильтра при $d(n) \in R^2$

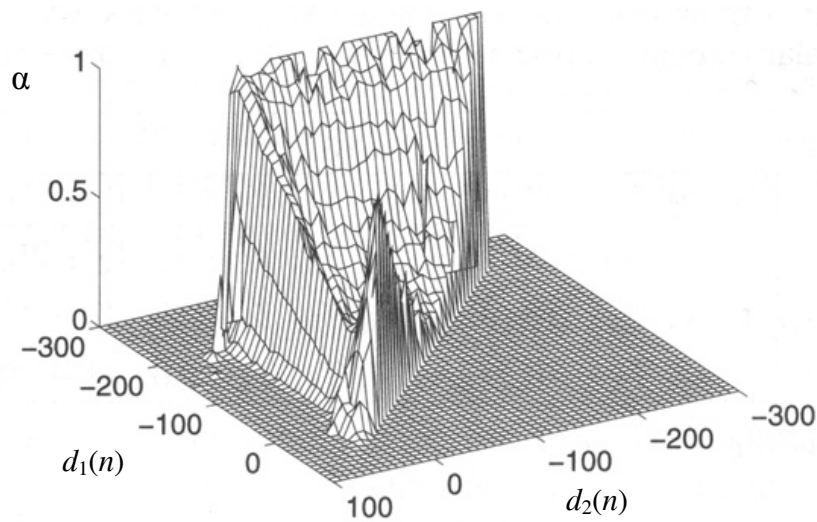


Рис. 4 Функция $\alpha(d(n))$ для обобщенного SD-ROM фильтра при $d(n) \in R^2$

При помощи нечеткой логики возможно лучшим образом обработать изображение в сравнении с пороговым SD-ROM фильтром, в отличие от которого обобщенный SD-ROM фильтр также эффективно удаляет гауссов

шум, а также гауссов и импульсный шум, которые одновременно воздействуют на изображение.

Значения коэффициентов α возможно получить с помощью оптимизационных алгоритмов. Полученные таким образом значения коэффициентов хранятся в памяти, а функция $\alpha(d(n))$ реализуется в виде таблицы. Но получаемая таким образом таблица имеет большой объем данных. Вследствие чего вычислительная сложность и требования к памяти становятся весьма существенными. Для решения данной задачи используется метод отображения четырехмерного векторного пространства R^4 в M неперекрывающихся областей A_i , $i = 1, 2, \dots, M$. После чего каждой области A_i ставится в соответствие определенное постоянное значение $\alpha(d(n))$. На рисунке 5 изображена данная реализация для $d(n) \in R^2$ и $M = 17^2$, при этом исходные коэффициенты изображены на рисунке 4.

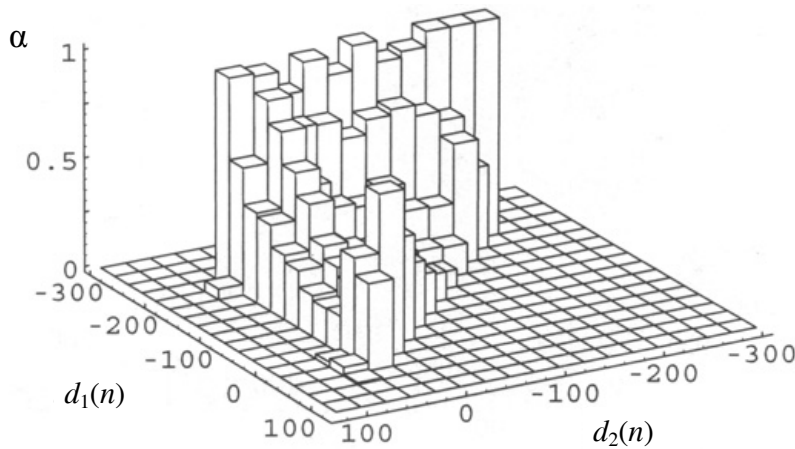


Рис. 5 Функция $\alpha(d(n))$ для обобщенного SD-ROM фильтра после аппроксимации

Таким образом, каждая область A_i определяется декартовым произведением четырех скалярных областей. Скалярная область определяется интервалом $(q_{i-1}, q_i]$, где q_i представляет собой порог, с помощью которого и осуществляется аппроксимация.

Обозначим α_i и β_i коэффициенты, которые получаются в результате аппроксимации коэффициентов $\alpha(d(n))$, $\beta(d(n))$ соответственно, т.е.

$$\alpha_i = \alpha(d(n)) \Big|_{d(n) \in A_i}, \quad (15)$$

и

$$\beta_i = \beta(d(n)) \Big|_{d(n) \in A_i}, \quad (16)$$

где $i = 1, 2, \dots, M$. При этом (13) примет вид

$$y(n) = \alpha_i x(n) + \beta_i m(n). \quad (17)$$

При реализации обычно используют $M = 6^4$ или $M = 8^4$, т.к. дальнейшее увеличение значения M несущественно влияет на конечный результат, на который главным образом воздействуют значения пороговых коэффициентов T .

В качестве оптимизационных алгоритмов используются алгоритмы наименьших квадратов и среднеквадратичной ошибки. В настоящей работе использовался алгоритм наименьших квадратов.

Результаты применения SD-ROM фильтров

На рисунке 6 показаны результаты работы медианных и SD-ROM фильтров при обработке изображения, подверженного импульсному шуму. Рассматриваемые изображения представляют собой изображения с 255 возможными уровнями серого. На рисунке 6,а дано исходное изображение, а на рисунке 6,б изображение, подверженное влиянию импульсного шума с вероятностью 0,2. Результат обработки простым медианным фильтром с маской 3×3 представлен на рисунке 6,в, где еще можно увидеть скопление шума. Рисунок 6,г демонстрирует работу порогового SD-ROM фильтра. По сравнению с медианным фильтром SD-ROM фильтр гораздо лучше справился с задачей удаления шума, т.к. визуально его практически не заметить. При использовании обобщенного SD-ROM фильтра (рис. 6,д) с $M = 6^4$ получаем изображение, которое еще ближе к оригиналу. В таблице 1 для результатов обработки приведены значения средней абсолютной ошибки (MAE), средней квадратичной ошибки (MSE) и значения пикового отношения сигнал/шум (PSNR), которое определяется:

$$PSNR = 10 \log_{10} \left(\frac{\sum_n 255^2}{\sum_n [y(n) - v(n)]^2} \right) dB, \quad (18)$$

где $v(n)$ – первоначальное изображение.

Таблица 1

Алгоритм	MAE	MSE	PSNR
Простой медианный фильтр с маской 3×3	4,31	90,4	28,6
Пороговый SD-ROM фильтр	1,32	29,22	33,5
Обобщенный SD-ROM фильтр с $M = 6^4$	1,33	22,30	34,6

Рассмотрим случай, когда изображение подвержено влиянию импульсного шума с вероятностью 0,4, т.е. почти каждый второй пиксель подвержен влиянию шума. Данное изображение приведено на рисунке 7,а.

Результат обработки изображения на рисунке 7,а простым медианным фильтром с маской 3×3 показан на рисунке 7,б. Результат обработки простым медианным фильтром с маской 5×5 изображен на рисунке 7,в. При сравнении результатов работы двух медианных фильтров видно, что фильтр с маской 5×5 лучше справился с задачей удаления шума, но существенно сгладил изображение.

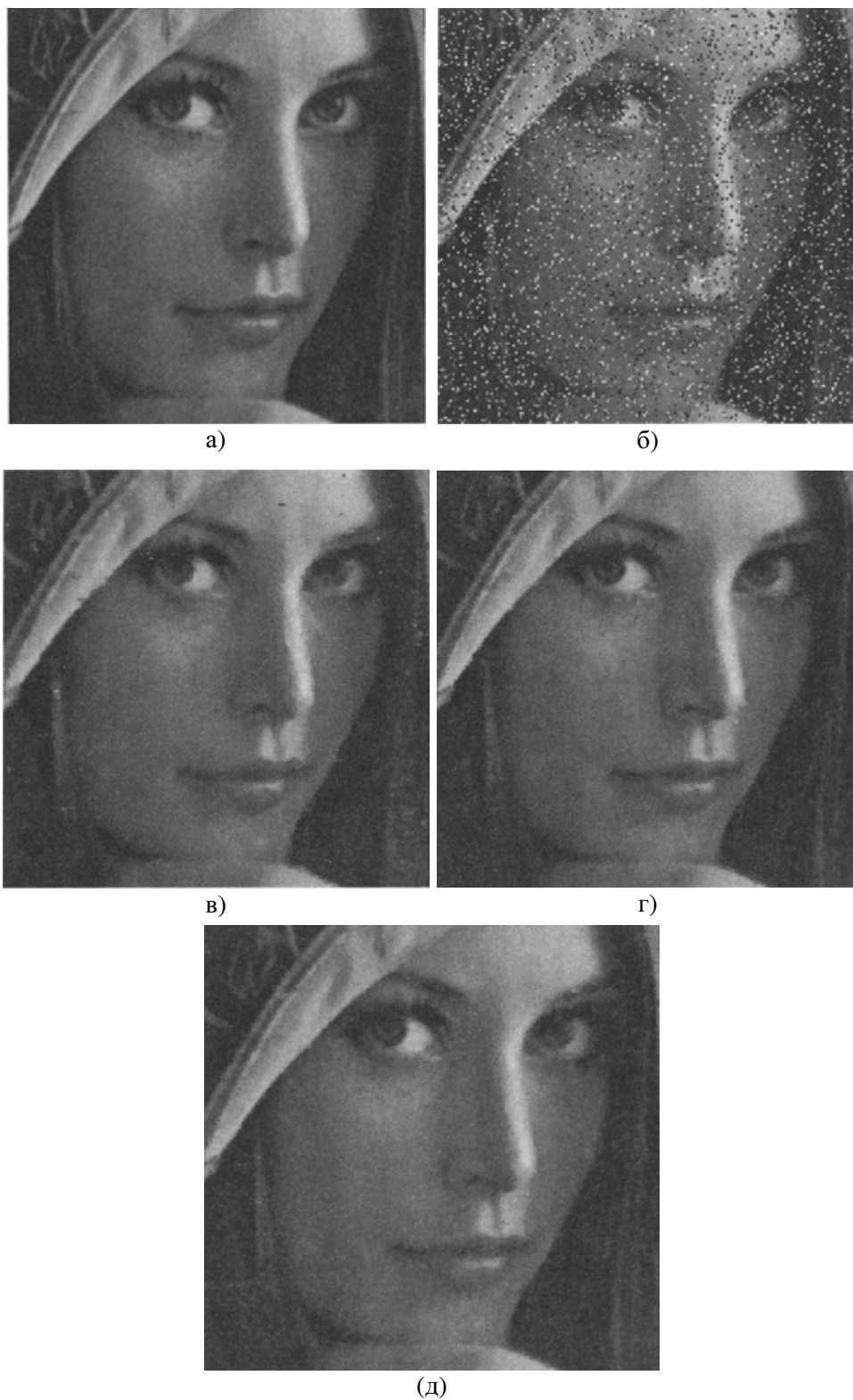


Рис. 6 Удаление импульсного шума при использовании медианных и SD-ROM фильтров

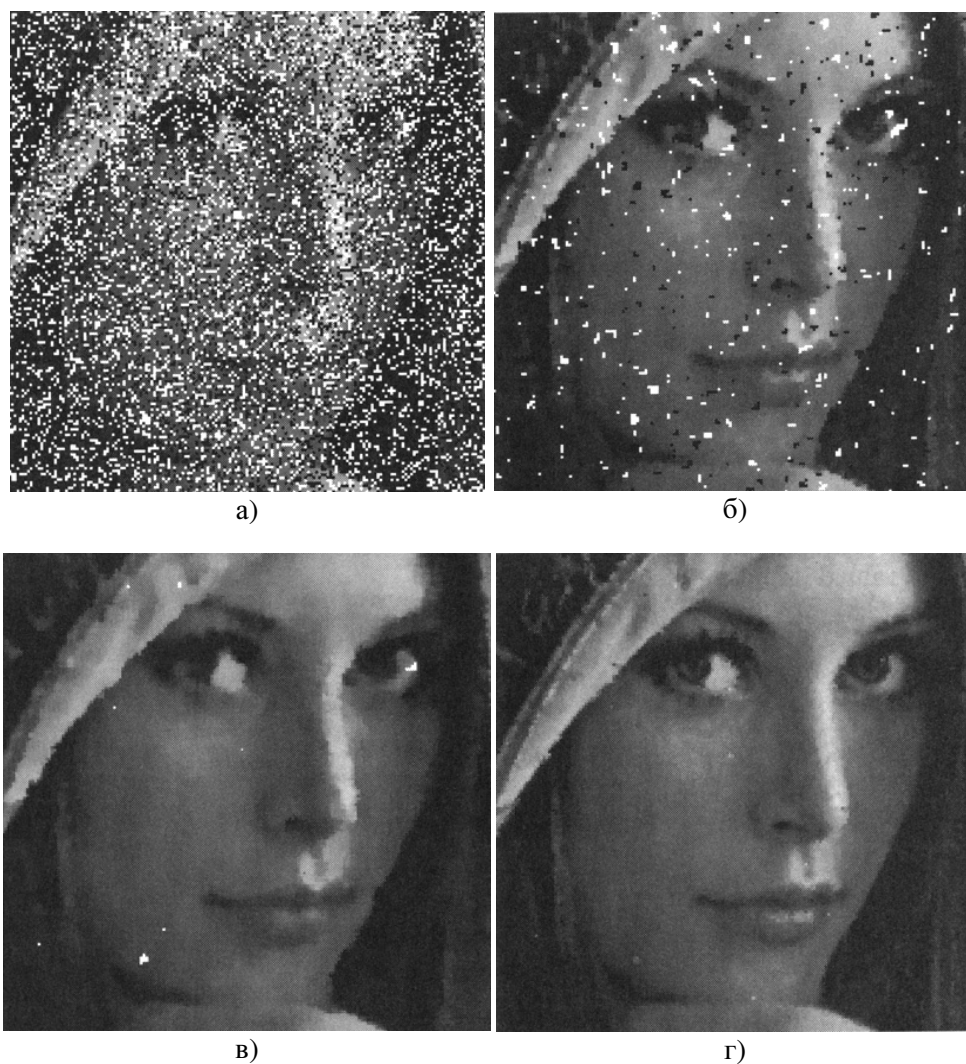


Рис. 7 Обработка изображения, подверженного влиянию 40% шума

Результат обработки пороговым SD-ROM фильтром показан на 7,г. Данный фильтр практически полностью удалил импульсный шум на изображении, при этом не наблюдается эффекта сглаживания изображения. В таблице 2 приведены значения средней абсолютной ошибки (MAE), средней квадратичной ошибки (MSE) и значения пикового отношения сигнал/шум (PSNR) для медианной и SD-ROM фильтрации, для изображений на рисунке 7.

Таблица 2

Алгоритм	MAE	MSE	PSNR
Простой медианный фильтр с маской 3×3	9,2	834,8	19,3
Простой медианный фильтр с маской 5×5	6,3	164,4	25,9
Пороговый SD-ROM фильтр	2,9	62,7	29,4

Заключение

Результаты исследования подтверждают тот факт, что на сегодня самым эффективным средством удаления импульсного шума считаются SD-ROM фильтры, которые также называют зависимыми от сигнала фильтрами среднего значения ранжированной последовательности. Они демонстрируют лучшие результаты в сравнении с другими медианными фильтрами, при этом пороговый SD-ROM фильтр позволяет практически полностью восстановить изображение, даже при уровне шума 40%. Реализация SD-ROM фильтров, в частности порогового, не требует больших вычислительных затрат с учетом существующих вычислительных мощностей.

Список литературы

1. **Сорокин, С. В.** Использование взвешенных медианных фильтров для удаления импульсного шума при обработке изображений / С. В. Сорокин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2007. – № 3. – С. 50–57.
2. **Abreu, E.** A signal dependent rank ordered mean (SD-ROM) filter – a new approach for removal of impulses from highly corrupted images / E. Abreu, S. K. Mitra // In Proc. Intl. Conf. on Acoustics, Speech, and Signal processing. – 1995. – V. 4. – P. 2371–2374.
3. **Abreu, E.** A new efficient approach for removal of impulse noise from highly corrupted images / E. Abreu, M. Lightstone, S. K. Mitra, K. Arakawa // IEEE Trans. Image Process., Special Issue on Nonlinear image processing. – 1996. – № 5 (6). – P. 1012–1025.
4. **Mitra, S.** Nonlinear image processing / S. Mitra, G. Sicuranza // Academic press. – 2001. – 455 p.
5. **Abreu, E.** A new efficient approach for removal of impulse noise from highly corrupted images / E. Abreu, M. Lightstone, S. K. Mitra, K. Arakawa // IEEE Trans. Image Process., Special Issue on Nonlinear image processing. – 1996. – № 5 (6). – P. 1012–1025.
6. **Florencio, D. A. F.** Decision-based median filter using local signal statistics / D. A. F. Florencio, R. W. Schafer // In Visual Communication and image processing, Proc. SPIE. – 1994. – V. 2308. – P. 268–275.
7. **Kundu, A.** Applications of two-dimensional generalized mean filtering for removal of impulse noise from images / A. Kundu, S. K. Mitra, P. P. Vaidyanathan // IEEE Trans. Acoust. Speech Signal Process. – 1984. – V. 32. – P. 600–609.
8. **Mitra, S. K.** A new nonlinear algorithm for removal of impulse noise from highly corrupted images / S. K. Mitra, T-H. Yu // IEEE Intl. Symp. On Circuits and Systems. – 1994. – V. 3. – P. 17–20.
9. **Sun, T.** Detail-preserving median based filters in image processing / T. Sun, Y. Neuvo // Patt. Recog. Lett. – 1994. – № 15. – P. 341–347.
10. **Chandra, C.** An efficient method for the removal of impulse noise from speech and audio signals / C. Chandra, M. S. Moore, S. K. Mitra // IEEE Intl. Symp. On Circuits and Systems. – 1998. – V. 4. – P. 206–209.
11. **Arakawa K.** A median filters based on fuzzy rules / K. Arakawa // IEICE Trans, J78-A(2). – 1995. – P. 123–131.