

УДК 004.92

DOI: 10.46960/1816-210X_2022_2_26

АДАПТИВНОЕ УПРАВЛЕНИЕ МЕДИАННОЙ ФИЛЬТРАЦИЕЙ ИЗОБРАЖЕНИЙ В MATHCAD

Е.А. НикулинORCID: 0000-0002-2520-681X e-mail: nea@mail.nnov.ruНижегородский государственный технический университет им. Р.Е. Алексеева
Нижний Новгород, Россия

Представлен обзор существующих в пакете MathCAD 15 средств генерации шумов и их фильтрации в растровых изображениях. Разработаны новые методы медианной фильтрации: с произвольными размерами прямоугольного окна и с оптимальными размерами квадратного окна, адаптивно подбираемыми в каждом пикселе изображения. Предложенные алгоритмы и реализующие их функции скомпонованы в законченные MathCAD-программы, очищающие изображения от различных помех с разными качественными показателями. Показана особая эффективность алгоритма адаптивной медианной фильтрации с настраиваемыми размерами окна.

Ключевые слова: изображение, шум, фильтрация, медиана, адаптивный алгоритм, MathCAD.

ДЛЯ ЦИТИРОВАНИЯ: Никулин, Е.А. Адаптивное управление медианной фильтрацией изображений в MathCAD // Труды НГТУ им. Р.Е. Алексеева. 2022. № 2. С. 26-35. DOI: 10.46960/1816-210X_2022_2_26

ADAPTIVE CONTROL OF MEDIAN FILTERING OF IMAGES IN MATHCAD

Е.А. NikulinORCID: 0000-0002-2520-681X e-mail: nea@mail.nnov.ruNizhny Novgorod state technical university n.a. R.E. Alekseev
Nizhny Novgorod, Russia

Abstract. Overview of existing in the MathCAD 15 package means of noise generation and filtering thereof in bitmap images, is presented. New median filtering methods are developed: with arbitrary rectangular window sizes and with optimal square window sizes adaptively selected in each pixel of the image. Proposed algorithms and functions implementing thereof are compiled into complete MathCAD programs that clean images from various interferences with different quality indicators. Special efficiency of the adaptive median filtering algorithm with adjustable window sizes, is shown.

Key words: image, noise, filtering, median, adaptive algorithm, MathCAD.

FOR CITATION: Nikulin E.A. Adaptive control of median filtering of images in MathCAD. Transactions of NNSTU n.a. R.E. Alekseev. 2022. № 2. Pp. 26-35. DOI: 10.46960/1816-210X_2022_2_26

Введение

Одной из важнейших задач компьютерной обработки изображений, представленных в виде матриц монохромных либо цветных пикселей, является фильтрация шумов, искажающих цветовые коды, в основном, случайным образом, по различным вероятностным законам. Помехи можно условно разделить на импульсные (искажены отдельные пиксели), линейные (царапины в виде цепочек пикселей) и площадные (сплошные пятна пикселей). Задача компьютерной фильтрации шумов в изображениях, основательно изученная в [1, 2], заключается в обнаружении и устранении этих помех с помощью алгоритмов и реализующих их программ. Наиболее популярным, но не самым эффективным, размывающим границы объектов, является линейный метод усреднения в маске, скользящей по матрице изображения.

Настоящая статья, посвященная решению задачи фильтрации в программной среде MathCAD [3], кратко знакомит с необходимыми встроенными функциями обработки изображений. Раскрываются некоторые эффекты их использования и более обстоятельно представляются усовершенствованные автором нелинейные алгоритмы медианной фильтрации, дающие подчас удивительные результаты очистки сильно зашумленных изображений. Моделью исследований служит монохромный фрагмент портрета знаменитой Джоконды.

Генерация шумов в MathCAD

Рассмотрим способы управляемого добавления пиксельных шумов в растровые изображения, после чего те могут служить основой для испытания методов шумоподавления в отсутствие реальных зашумленных рисунков.

Самой первой в категории «Обработка изображений» панели «Вставка функции» стоит функция $\text{addnoise}(M, p, s)$, которая с равными вероятностями $p/2$ увеличивает либо уменьшает код каждого элемента матрицы M точно на s единиц. При выходе измененного кода из байтового интервала $\Delta = [0, 255]$ он самой функцией автоматически сдвигается на его ближайшую границу, тем самым исключая сдвиг кодов шаблоном рисунка в Δ на ± 256 . Добавление импульсного шума с разными вероятностями p и фиксированными смещениями s реализовано в верхнем ряду изображений на рис. 1. Они демонстрируют уплотнение импульсных помех с ростом вероятности их появления в половине пикселей при $p=0.5$ и в каждом при $2p=1$, а также повышение контраста зашумленных точек с окружающим фоном при увеличении параметра разброса от $s=50$ до удвоенного $2s=100$.

В отсутствие в MathCAD других встроенных функций зашумления матричных изображений создадим пользовательские функции:

$$\text{unif}(c, s) := \text{runif}(1, c - s, c + s)_0, \quad \text{norm}(c, s) := \text{rnorm}(1, c, s)_0 \quad (1)$$

переменных смещений числа c , использующие встроенные функции генерирования случайных векторов: runif для равномерно распределенных элементов векторов в интервале $(c-s, c+s)$ и rnorm для нормально распределенных элементов с матожиданием (МО), равным c , и среднеквадратичным отклонением (СКО), равным s . По свойству нормального распределения 99,73 % смещенных чисел лежат вокруг значения c в интервале $(c-3s, c+3s)$.

Анализ изображений во втором ряду рис. 1 подтверждает вполне ожидаемые эффекты расположения случайных пиксельных шумов и формирования их интенсивностей:

- плотности равномерно и нормально распределенных пиксельных помех при одинаковых значениях параметра разброса s заметно отличаются друг от друга из-за практически втрое большей ширины интервала расположения гауссовских случайных чисел в сравнении с интервалом локации равномерных;
- цвет большинства точечных помех в матрицах, полученных векторизацией функций (1) от матрицы M , оказался противоположным цвету окружающих фоновых пикселей по принципу «соль и перец» — светлые точки на темном фоне и темные — на светлом.

При этом с удвоением параметра разброса s такой контрастный шум почти полностью «забил» исходное изображение. В реальности выводимые матрицы таких помех не имеют, а «перец» на светлом челе Моны Лизы и белая «соль» на ее темных волосах — всего лишь иллюзия. Такие эффекты инверсии цвета пикселей объясняются тем, что в них зашумленные матрицы имеют внебайтовые коды, которые шаблоном рисунка лишь *визуально* конвертируются в интервал Δ путем параллельного сдвига на ± 256 .

Поручим приведение случайных чисел, сгенерированных в (1), в байтовый диапазон Δ следующим функциям:

$$\text{uni}(c, s) := \min(\max(\text{unif}(c, s), 0), 255), \quad \text{nor}(c, s) := \min(\max(\text{norm}(c, s), 0), 255). \quad (2)$$

В нижнем ряду рис. 1 в изображениях, зашумленных функциями (2), эффекты конвертации, присущие шаблону вывода рисунка, исчезли.

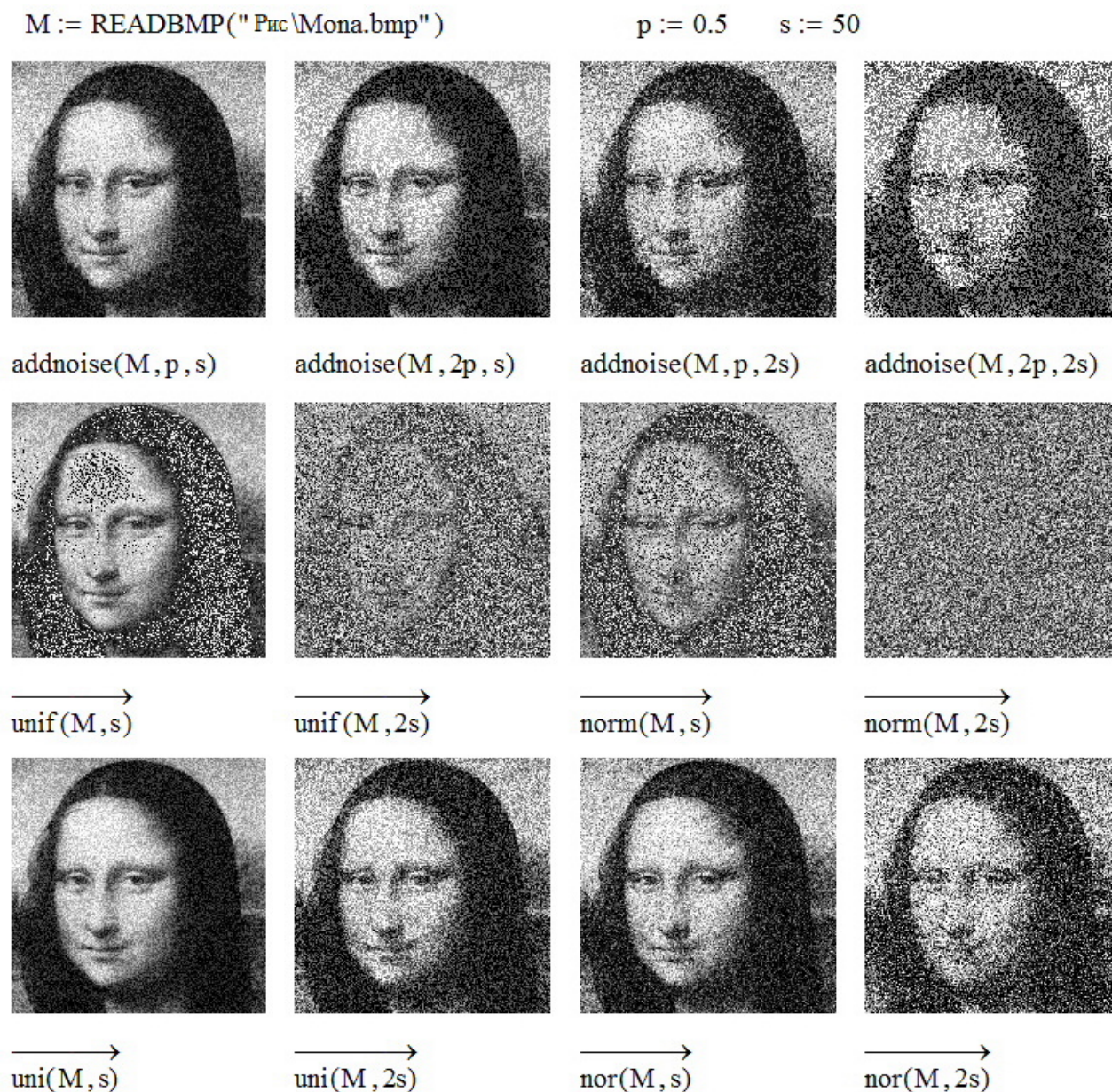


Рис. 1. Импульсный шум в изображениях

Fig. 1. Impulse noise in images

Функции линейной фильтрации изображений в MathCAD

Переходим к решению задачи фильтрации шума с помощью встроенных функций MathCAD, собранных в категории «Обработка изображений» панели «Вставка функции» и предназначенных для *сглаживания* импульсных помех в монохромных изображениях. Методы подавления помех разделяются на пространственные и частотные [1, 2], но из-за ограниченности возможностей протестировать здесь все функции будут рассмотрены лишь некоторые из них, выполняющие *пространственную* фильтрацию.

Как в радиотехнике для устранения пульсаций сигнал пропускается через фильтр низких частот, так и в обработке изображений, зашумленных пиксельными помехами, используется обладающий интегрирующим свойством *метод усреднения* яркостей пикселей в окружающем обрабатываемую точку прямоугольном *окне* (ядре, маске, апертуре) $W \in \mathbb{R}^{H \times D}$ *нечетных* размеров H и D с полями шириной сверху и снизу по $h=(H-1)/2$, а слева и справа по

$d=(D-1)/2$. Общее свойство всех масок, сохраняющее яркости пикселей однотонного фона, – сумма всех элементов W_{yx} равна единице. Яркость (i,j) -го пикселя матрицы $M \in \mathbb{R}^{(m+1) \times (n+1)}$ в окне $W \in \mathbb{R}^{(2h+1) \times (2d+1)}$, центрированном в элементе M_{ij} , рассчитывается по формуле:

$$S_{ij} = \sum_{y=\max\{-h,-i\}}^{\min\{h,m-i\}} \sum_{x=\max\{-d,-j\}}^{\min\{d,n-j\}} W_{yx} M_{i+y,j+x} \quad \forall i \in [0,m], j \in [0,n]. \quad (3)$$

Результирующая матрица свертки S имеет такие же размеры, как и матрица M . Указанные в (3) пределы сумм предотвращают при неполном покрытии окна выход индексов элементов на краях матрицы M за пределы интервалов $[0,m]$ и $[0,n]$, обеспечивая выполнение неравенств $0 \leq i+y \leq m$ и $0 \leq j+x \leq n$. При полном покрытии матрицей окна $y=-h\dots h$ и $x=-d\dots d$.

На рис. 2 продемонстрирована работа имеющихся в MathCAD функций сглаживания со встроенными в них масками

$$W_3 = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad W_5 = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad W_c = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad W_{03} = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \quad W_{05} = \frac{1}{32} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & 4 & 2 & 1 \\ 1 & 1 & 2 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} :$$

- $\text{unismooth}(M)$ и $\text{unismooth5}(M)$ с однородными масками W_3 и W_5 ;
- $\text{centsmooth}(M)$ с центрально-взвешенным 3×3 -окном W_c , вносящим вдвое больший вес цвета сглаживаемого пикселя M_{ij} в общую сумму (3) с целью уменьшения размытия границ полезных фигур изображения;
- $\text{orthosmooth}(M)$ и $\text{orthosmooth5}(M)$ с ортогонально-взвешенными масками W_{03} и W_{05} , также уменьшающими размытие и расфокусировку кромок фигур в сравнении с однородными окнами тех же размеров.

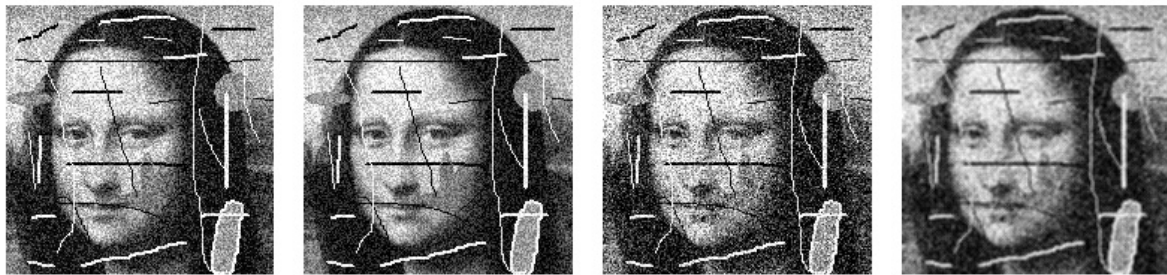
В нижнем ряду рис. 2 показано применение функций $\text{convol2d}(M,W)$, $\text{convolve3}(M,W_{c3})$ и $\text{convolve5}(M,W_{c5})$ фильтрации с внешне задаваемыми окнами: $W \in \mathbb{R}^{H \times D}$ произвольных размеров $\{H, D\}$ и квадратными масками фиксированных размеров $W_{c3} \in \mathbb{R}^{3 \times 3}$ и $W_{c5} \in \mathbb{R}^{5 \times 5}$.

Работа перечисленных функций оконной фильтрации продемонстрирована на примере изображения $M = \langle \text{Царап.bmp} \rangle$, созданного в графическом редакторе путем нанесения на рисунок $\langle \text{Мона.bmp} \rangle$ линейных (царапин разных толщин и направлений) и площадных (пятен разных размеров и форм) помех. Последующие добавления в матрицу M пиксельных шумов функциями $\text{addnoise}(M,0.5,s)$, $\text{uni}(M,s)$ и $\text{nor}(M,s)$ с параметром разброса $s=50$ сформировали матрицы Ma , Mu и Mn . Анализ этих изображений и результатов их фильтрации привел к следующим выводам:

- из трех типов пиксельного шума наименее плотным является фиксированный, равновероятно добавленный в матрице Ma к элементам M_{ij} как 0 либо ± 50 ; из оставшихся менее зернистым и контрастным выглядит равномерно распределенный в интервале $(-50,50)$ шум в матрице Mu , а наиболее контрастным – нормально распределенный с 99,73 % интервалом $(-150,150)$ гауссовский шум в матрице Mn ;
- все методы оконного сглаживания обладают присущим им свойством *размытия* как отдельных пикселей, так и границ широких царапин и пятен, расширяющим резкие перепады яркостей в пологие блеклые площадки и полосы; размытие границ проявляется тем сильнее, чем больше размеры окна фильтрации;
- с увеличением размера окна появляется и далее расширяется черная окантовка изображения, что объясняется меньшим количеством усредняемых пикселей, накрытых окном на краях рисунка;
- в нижнем ряду рис. 2 показана возможность применения не встроенных в функции свертки, а внешне задаваемых масок. Однострочное $W = [1 \ 2 \ 4 \ 8 \ 4 \ 2 \ 1]/22$ и одностолбцовое W^T окна избирательно сглаживают функцией convol2d вертикальные и горизонтальные

тонкие царапины. Там же видно действие двух квадратных окон: крестовидного $W3$ и квадратного $W5$, усредняющего 24 числа, окружающих M_{ij} .

```
M := READBMP("Рис.Царап.bmp")
s := 50  Ma := addnoise(M, 0.5, s)
Mu := uni(M, s)  Mn := nor(M, s)
W := (1 2 4 8 4 2 1) ÷ 22
W3 := 1/6 * [ 0 1 0
              1 2 1
              0 1 0 ]
W5 := 1/24 * [ 1 1 1 1 1
                1 1 1 1 1
                1 1 0 1 1
                1 1 1 1 1
                1 1 1 1 1 ]
```



Ma Mu Mn centsmooth(Mn)



unismooth(Ma) unismooth(Mu) unismooth(Mn) orthosmooth(Mn)



unismooth5(Ma) unismooth5(Mu) unismooth5(Mn) orthosmooth5(Mn)



convol2d(Mn, W) convol2d(Mn, W^T) convolve3(Mn, W3) convolve5(Mn, W5)

Рис. 2. Встроенные функции линейной фильтрации

Fig. 2. Built-in Line Filtering Functions

Медианная фильтрация изображений

Возможности оконной фильтрации изображений не исчерпываются линейным алгоритмом (3). Среди других, *нелинейных* операций с яркостями пикселей, попавших в скользящее по рисунку окно, можно назвать выборку минимального, максимального, средних квадратичного, геометрического, гармонического и других математических величин, например, *медианы* – срединного по расположению числа в упорядоченной по возрастанию последовательности чисел.

Работа встроенного *медианного фильтра* $\text{medfilt}(M)$, предназначенного для очистки монохромных изображений M от импульсных (пиксельных) помех, основана на поочередном совмещении центра квадратного окна фиксированного размера 3×3 с каждым пикселем рисунка M_{ij} , сортировке яркостей покрытых окном пикселей и запоминании в элементе возвращаемой матрицы S_{ij} медианы этой последовательности.

Желая реализовать свойства ориентированной (горизонтальной или вертикальной) линейной фильтрации, присущей функции convol2d , создадим универсальный аналог медианного фильтра матрицы M с окном произвольных размеров $H \times D$, где H и D — *нечетные* (1, 3, 5, 7, ...) габариты окна с полями вокруг центральной точки шириной сверху и снизу $h=(H-1)/2$, а слева и справа $d=(D-1)/2$ пикселей. В приведенной ниже функции $\text{medfilW } H \times D$ -окно, сканирующее все пиксели матрицы M , можно задавать не только квадратным, но и как в функции convol2d , *прямоугольным*, в том числе, однострочным ($H=1, h=0$) или одностолбцовым ($D=1, d=0$). Это позволяет фильтровать не только импульсные, но и координатно ориентированные *сплошные* протяженные помехи типа царапин и трещин, а именно: горизонтальные царапины рекомендуется сглаживать вертикальными $H \times 1$ -масками, а вертикальные – горизонтальными масками размерами $1 \times D$.

```

medfilW(M, H, D) := | "Медианный фильтр в окне размером H*D"
                    | [ n ← cols(M) - 1  h ← (H - 1) ÷ 2  d ← (D - 1) ÷ 2 ]
                    | for i ∈ 0..m ← rows(M) - 1
                    |   | ( y ← max(0, i - h)  Y ← min(m, i + h) )
                    |   | for j ∈ 0..n
                    |   |   | ( x ← max(0, j - d)  X ← min(n, j + d) )
                    |   |   | Si,j ← median(submatrix(M, y, Y, x, X))
                    | S

```

Сверху и справа на рис. 3 показаны изображение N , полученное импульсным нормально распределенным зашумлением поцарапанного рисунка $M=\langle\text{Царап.bmp}\rangle$ функцией $\text{nor}(M,s)$ с СКО $s=50$ и результаты его медианной фильтрации MathCAD-функцией medfilt в окне 3×3 и универсальной функцией medfilW в квадратных окнах $3 \times 3 \div 9 \times 9$. Очевидно, что в 3×3 -окне обе функции работают одинаково. Во втором и третьем рядах сконструированная функция medfilW испытана на неквадратных окнах.

Сопоставление размеров, плотности, яркости и контраста оставшихся на изображениях шероховатостей позволяет сделать новые выводы и подтвердить ранее сделанные предсказания о свойствах медианной фильтрации монохромных изображений окнами разных размеров и конфигураций:

- импульсные (точечные) помехи расплылись в пятна, размер которых примерно совпадает с размером окна фильтрации; что касается наблюдаемого контраста пиксельных помех с окружающим фоном, с ростом размера окна от 3×3 до 9×9 он неуклонно уменьшается вплоть до еле различимого;



Рис. 3. Универсальная медианная фильтрация

Fig. 3. Universal Median Filtering

- это же свойство размытия точечных помех приводит к расплыванию и ухудшению четкости границ не только площадных шумов (пятен), но и полезных фигур, составляющих сюжет изображения; при этом наиболее сильно размываются границы, ориентированные вдоль максимального размера окна;
- медианная фильтрация справляется с возложенной на нее задачей качественнее, чем функции линейного сглаживания, работа которых показана на рис. 2; особенно наглядно ее преимущества проявляются при устранении шумов в виде отдельных точек и линий толщиной менее половины максимального размера окна (после сортировки экстремальные яркости пикселей окна группируются на краях числовой последовательности («перец» в начале, «соль» в конце), оставляя в середине (медиане) яркости фоновых пикселей, благодаря чему контраст зашумленного пиксела в центре окна существенно понижается);
- горизонтальные царапины хорошо сглаживаются вертикально вытянутыми окнами фильтра, а вертикальные – горизонтальными окнами; наклонные и толстые штрихи следует устранять медианным фильтром с квадратным окном;
- помехи в форме пятен размерами более, чем $\max\{H, D\}$, фильтруются плохо, сглаживаются только их контрастные границы.

Адаптивная медианная фильтрация изображений

Приведенные примеры медианной фильтрации зашумленного изображения M как встроенной функцией $\text{medfilt}(M)$ с фиксированным 3×3 -окном, так и универсальной функцией $\text{medfilW}(M, H, D)$ с окнами произвольных размеров $H \times D$ демонстрируют разное качество удаления шумов различного типа и ставят закономерные вопросы о выборе оптимального размера окна и правомочности применения одного и того же размера на всех участках изображения, отличающихся разными яркостями, плотностями и типами помех. Ответом на эти вопросы является усовершенствование предложенного в [1] алгоритма *адаптивной медианной фильтрации* изображения $M \in \mathbb{R}^{(m+1) \times (n+1)}$, подбирающего в каждом пикселе M_{ij} с индексами $i \in [0, m]$ и $j \in [0, n]$ наиболее подходящий размер квадратного $(2d+1) \times (2d+1)$ -окна, начиная с минимальной ширины всех полей вокруг его центральной точки $d=1$. Перед алгоритмом ставятся следующие цели:

- удалить бинарные (черные и белые) импульсные помехи и максимально сгладить шумы с другими функциями распределения (равномерными, нормальными и т. п.);
- минимизировать размытие границ протяженных объектов, составляющих основу сюжета изображения.

Реализующая эти задачи функция

```
medfilA(M) := | "Адаптивный медианный фильтр"
               | D ← floor(min(m ← rows(M) - 1, n ← cols(M) - 1) ÷ 2)
               | for i ∈ a ← 0..m
               |   for j ∈ 0..n
               |     for d ∈ 1..D
               |       | ( y ← max(f ← 0, i - d)  Y ← min(m, i + d) )
               |       | O ← submatrix(M, y, Y, max(0, j - d), min(n, j + d))
               |       | ( mi ← min(O)  me ← median(O)  ma ← max(O) )
               |       | ( f ← 1  break ) if mi < me < ma
               |       | [ Ai,j ← if[(mi < Mi,j < ma) · f, Mi,j, me]  a ← max(a, d) ]
               | ( A 2a + 1 )
```

настраивает во вложенном цикле $i=0, m, j=0, n$ для каждого пиксела изображения M_{ij} его адаптивную медианную яркость A_{ij} , руководствуясь следующей логикой:

- во внутреннем цикле $d=1, D$, где $D = \lfloor \min\{m, n\} / 2 \rfloor$, производится поиск ширины поля $d \leq D$, при которой размер окна $(2d+1) \times (2d+1)$ не превышает минимального размера изображения $2D+1$. Для этого из матрицы M выбирается прямоугольный блок O покрытых окном пикселей и вычисляются их экстремальные $mi = \min\{O\}$, $ma = \max\{O\}$ и медианное $me = \text{median}\{O\}$ значения яркостей;
- при выполнении неравенств $mi < me < ma$ бинарные шумы, попавшие в окно, сгруппируются на краях упорядоченной последовательности $\text{sort}(O)$ – «перец» на левом, а «соль» на правом. Если при этом еще и $mi < M_{ij} < ma$, то центр окна также не будет засорен бинарным шумом, что позволяет сохранить его яркость M_{ij} в (i, j) -м элементе возвращаемой матрицы $A_{ij} = M_{ij}$ и минимизировать искажения, вносимые обработкой изображения. В противном случае при $mi = M_{ij}$ либо $M_{ij} = ma$ сохраняется менее зашумленное медианное значение $A_{ij} = me$, соответствующее обычному медианному фильтру;
- если же $mi = me$ либо $me = ma$, то экстремальные яркости заполнили не менее половины окна O . Это бывает при большой плотности шумов и недостаточном значении d . После инкрементирования $d++$ выполнение предыдущего шага повторяется до тех пор, пока $d \leq D$;

- по окончании цикла подбора величины d элементу матрицы A_{ij} присваивается значение медианы me как наиболее компромиссного результата фильтрации.

Функция $medfilA(M)$, работа которой иллюстрируется на рис. 4, возвращает список, включающий матрицу адаптивно-медианных яркостей A и максимально достигнутый размер $a=2d_{max}+1$ квадратного $a \times a$ -окна, где d_{max} – максимальный размер поля, полученный во всех циклах подбора значений d для всех пикселей изображения M_{ij} .

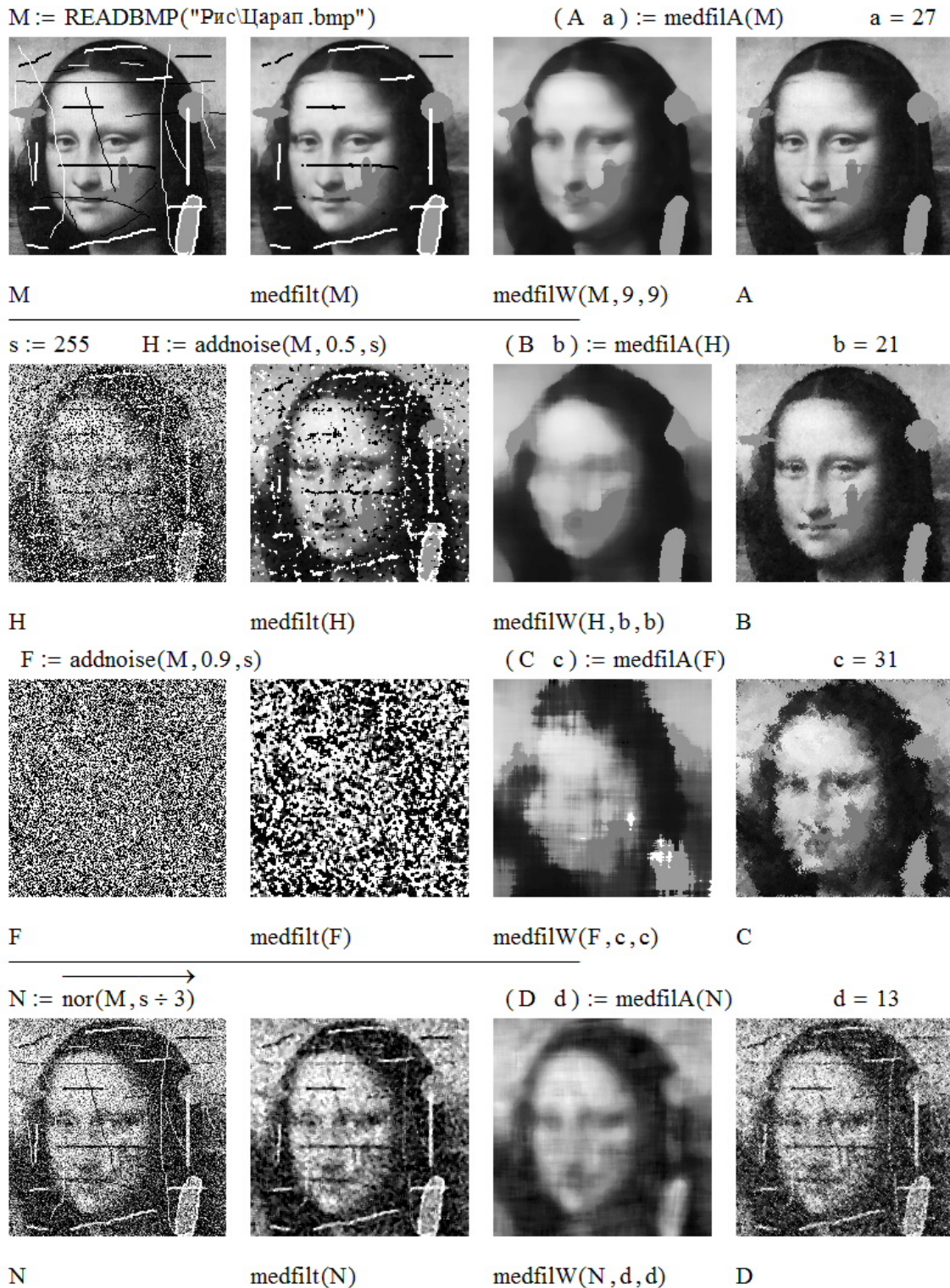


Рис. 4. Сравнение функций медианной фильтрации

Fig. 4. Comparison of median filtering functions

В представленной программе проведено сравнение трех медианных алгоритмов фильтрации изображения M =«Царап.bmp», содержащего царапины и пятна разных размеров и яркостей. В верхнем ряду встроенный фильтр $medfilt(M)$ смог удалить своим 3×3 -окном только тонкие, толщиной в один пиксел, царапины, тогда как адаптивный фильтр $medfilA(M)$ с максимальной шириной окна $a=27$ удалил их все, сохранив при этом четкость границ. Для сравнения алгоритм $medfilW(M,9,9)$ с вручную подобранными размерами окна 9×9 , удалившими следы царапин, сильно размыл контуры фигуры. В двух центральных рядах к изображению M добавлены импульсные шумы $addnoise(M,p,s)$ с параметром $s=255$, делающим их бинарными. При вероятности $p=0.5$ функция $medfilA(H)$ с максимальным размером окна $b=21$ удалила и их, и царапины полностью и без размытия границ сплошных объектов, чего не смог сделать алгоритм $medfilW(H,b,b)$ с фиксированным $b \times b$ -окном. И даже при 90 % плотности импульсного шума, полностью скрывшего исходный рисунок в хаосе изображения F , алгоритм $medfilA(F)$ с максимальным размером окна $c=31$ смог более-менее «вытянуть» знакомый образ по сравнению с сильно смазанным изображением $medfilW(F,c,c)$, полученным при постоянном размере окна $c \times c$. Наконец, в нижнем ряду показаны весьма посредственные результаты фильтрации, в том числе, и адаптивной, гауссовского шума в изображении N .

Выводы

Сравнение встроенных в MathCAD функций фильтрации шумов в растровых изображениях и вновь разработанных алгоритмов и функций медианной фильтрации однозначно показало более высокое качество очистки последними разнообразных помех – импульсных, линейных и сплошных.

Библиографический список

1. **Гонсалес, Р.** Цифровая обработка изображений [Текст] / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2012. – 1104 с.
2. **Яне, Б.** Цифровая обработка изображений [Текст] / Б. Яне. – М.: Техносфера, 2007. – 584 с.
3. **Кириянов, Д.В.** Mathcad 15 / Mathcad Prime 1/0 [Текст] / Д.В. Кириянов. – СПб.: БХВ-Петербург, 2012. – 432 с.

*Дата поступления
в редакцию: 14.03.2022*