

УДК 004.92

DOI: 10.46960/1816-210X\_2023\_1\_21

## АНАЛИЗ МЕТОДОВ ПОВЫШЕНИЯ ЧЕТКОСТИ И ОБНАРУЖЕНИЯ ГРАНИЦ ИЗОБРАЖЕНИЙ В MATHCAD

Е.А. Никулин

ORCID: 0000-0002-2520-681X e-mail: nea@mail.nnov.ru

Нижегородский государственный технический университет им. Р.Е. Алексеева  
Нижний Новгород, Россия

Представлен обзор существующих в пакете MathCAD 15 средств повышения четкости низкоконтрастных, размытых и зашумленных растровых изображений, а также обнаружения границ между их объектами и фоном. На ряде монохромных изображений выполнены разностороннее испытание встроенных функций и подбор их оптимальных параметров, показана возможность их совместного использования с другими функциями, повышающими качество восприятия изображений. Доказана особая эффективность методов Кэнни, эрозии и дилатации в задаче обнаружения границ объектов.

**Ключевые слова:** изображение, четкость, граница, градиент, лапласиан, MathCAD.

**ДЛЯ ЦИТИРОВАНИЯ:** Никулин, Е.А. Анализ методов повышения четкости и обнаружения границ изображений в MathCAD // Труды НГТУ им. Р.Е. Алексеева. 2023. № 1. С. 21-32. DOI: 10.46960/1816-210X\_2023\_1\_21

## ANALYSIS OF METHODS OF ENHANCEMENT AND DETECTING BORDERS OF IMAGES IN MATHCAD

E.A. Nikulin

ORCID: 0000-0002-2520-681X e-mail: nea@mail.nnov.ru

Nizhny Novgorod state technical university n.a. R.E. Alekseev  
Nizhny Novgorod, Russia

**Abstract.** Presented is review of tools for enhancement of low-contrast, unsharp and noisy bitmap images and for detecting borders between objects and background on images. For a series of monochrome images, multifaceted testing of embedded functions and their optimal adjustments are performed and the probability of their combined use with other image enhancement functions is shown. Canny methods, erosion and dilation demonstrate particular effectiveness for the task of object border detecting.

**Key words:** image, sharpness, border, gradient, Laplacian, MathCAD.

**FOR CITATION:** Nikulin E.A. Analysis of methods of enhancement and detecting borders of images in MathCAD. Transactions of NNSTU n.a. R.E. Alekseev. 2023. № 1. Pp. 21-32. DOI: 10.46960/1816-210X\_2023\_1\_21

### Введение

Известно, что фильтрация шумов растровых изображений разными методами сопровождается, к сожалению, большим или меньшим размытием границ площадных объектов, т.е. расширением участков перепадов яркостей объектов и фона и уменьшением их контраста [1, 2]. Даже описанный в [3] усовершенствованный метод адаптивной медианной фильтрации не лишен отмеченных недостатков. После сглаживания помех представляется целесообразным по возможности восстановить четкость границ объектов: сузить их ширину и увеличить перепады яркостей. Более наглядный способ выделения границ заключается в их изображении белыми линиями на черном фоне либо негативно – черными линиями на белом фоне. В статье будет рассмотрено решение этих задач в программной среде MathCAD, наби-

рающей популярности благодаря компактности записи формул максимально близко к естественному математическому языку, простоте построения всех видов графиков и способности к символическим вычислениям [4].

Общим у методов повышения четкости и обнаружения границ объектов растрового монохромного (в градациях серого) изображения, пиксели которого составляют матрицу  $M \in \mathbb{R}^{(m+1) \times (n+1)}$ , является вычисление яркости  $(i, j)$ -го пиксела  $M_{ij}$  по формуле *свертки*:

$$S_{ij} = \sum_{y=\max\{-h,-i\}}^{\min\{h,m-i\}} \sum_{x=\max\{-d,-j\}}^{\min\{d,n-j\}} W_{yx} M_{i+y,j+x} \quad \forall i \in [0,m], j \in [0,n]. \quad (1)$$

Здесь  $W \in \mathbb{R}^{H \times D}$  – окружающее этот пиксел прямоугольное *окно* нечетных размеров  $H$  и  $D$  с полями шириной сверху и снизу по  $h = (H-1)/2$ , а слева и справа по  $d = (D-1)/2$ . Результирующая матрица свертки  $S$  имеет размеры матрицы  $M$ . Указанные в (1) 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  $x = -d \dots d$ .

Представленные в статье MathCAD-программы работают со следующими изображениями, читаемыми из файлов \*.bmp встроенной функцией READBMP("\*.bmp") (рис. 1):

- $M$  из файла Mona.bmp, содержащего монохромный фрагмент известной картины с яркостями пикселей в полнобайтовом интервале  $[0,255]$ ;
- $C$  из файла Царап.bmp – предыдущее изображение с намеренно нанесенными на него линейными (царапинами разных толщин и направлений) и площадными (пятнами разных размеров и форм) помехами;
- $B$  из файла Мозг.bmp – фото среза мозга, изобилующее множеством мелких мышц и извилин серого вещества;
- $P$  из файла Палец.bmp – низкоконтрастный темный отпечаток пальца с яркостями пикселей в интервале  $[37,183]$ ;
- $A$  из файла Air.bmp – низкоконтрастный засвеченный снимок аэропорта с яркостями пикселей в интервале  $[30,255]$ .

```
M := READBMP("Mona.bmp")   B := READBMP("Мозг.bmp")   P := READBMP("Палец.bmp")
min(M) = 0   max(M) = 255   min(B) = 0   max(B) = 255   min(P) = 37   max(P) = 183
C := READBMP("Царап.bmp")   A := READBMP("Air.bmp")   min(A) = 30   max(A) = 255
```



M

C

B

P

A

Рис. 1. Исходные изображения

Fig. 1. Raw images

### Повышение четкости границ объектов

В багаже инструментов MathCAD есть набор функций повышения четкости границ объектов монохромного изображения  $M$  путем вычисления по (1) его свертки со встроенными окнами:

$$W_u = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \quad W_d = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}, \quad W_{o3} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad W_{o5} = \begin{bmatrix} 0 & -1 & 1 & -1 & 0 \\ -1 & 2 & -4 & 2 & -1 \\ 1 & -4 & 13 & -4 & 1 \\ -1 & 2 & -4 & 2 & -1 \\ 0 & -1 & 1 & -1 & 0 \end{bmatrix}; \quad (2)$$

- $\text{unicrisp}(M)$  с равномерно-взвешенным  $3 \times 3$ -окном  $W_u$ ;
- $\text{diacrisp}(M)$  с диагонально-взвешенным  $3 \times 3$ -окном  $W_d$ ;
- $\text{orthocrisp}(M)$  и  $\text{orthocrisp5}(M)$  с ортогонально-взвешенными окнами  $W_{o3} \in \mathbb{R}^{3 \times 3}$  и  $W_{o5} \in \mathbb{R}^{5 \times 5}$ .

Благодаря отрицательным весам в матрицах окон (2) сумма (1) содержит в  $(i,j)$ -м пикселе, совмещенном с центром окна  $W_{0,0}$ , разности яркостей  $M_{ij}$  и соседних пикселей. Так, для окна  $W_{o3}$  получим:

$$S_{ij} = M_{ij} + (M_{ij} - M_{i,j-1}) + (M_{ij} - M_{i,j+1}) + (M_{ij} - M_{i-1,j}) + (M_{ij} - M_{i+1,j}).$$

В результате операция свертки приобретает дифференцирующие свойства, позволяющие обнаружить резкие перепады яркостей в граничных пикселях  $M_{ij}$ , где значение  $S_{ij}$  сильно изменяется (вплоть до выхода за оба предела байтового интервала  $\Delta = [0,255]$ ) по сравнению с пикселями, расположенными в фоновой части рисунка. Побочным эффектом повышения резкости размытых границ является дифференцирование малых флуктуаций яркости фоновых пикселей, оставшихся после сглаживания импульсных шумов, из-за чего на однородных участках изображения могут появиться артефакты в виде мелких чипсов (*crisps*). Работа перечисленных функций продемонстрирована в программе на рис. 2, где изображение *Mona.bmp* было преднамеренно наполовину затемнено до  $M2 \in [0,128]$  и размыто встроенным медианным фильтром *medfilt* с окном размерами  $3 \times 3$ . Далее это сглаженное с небольшим понижением контраста изображение  $F \in [4,123]$  было обработано четырьмя фильтрами повышения четкости с результирующими матрицами  $U = \text{unicrisp}(F)$ ,  $D = \text{diacrisp}(F)$ ,  $O = \text{orthocrisp}(F)$  и  $O5 = \text{orthocrisp5}(F)$ . Анализ полученных результатов привел к следующим наблюдениям:

- все испытанные функции дали достаточно темные низкоконтрастные изображения, в которых большие значения центральных элементов матриц окон (2) обусловили выход некоторых яркостей в матрицах  $U \in [-164,531]$ ,  $D \in [-55,179]$ ,  $O \in [-54,266]$  и  $O5 \in [-202,621]$  за пределы байтового интервала  $\Delta$ ; вследствие этого на изображениях появились *ложные* белые и черные точки и линии, сдвинутые шаблоном рисунка в  $\Delta$  на  $\pm 256k$ ;
- для удаления внебайтовых кодов из содержащих их матриц можно предпринять следующие действия:
  - 1) обрезать их функцией  $\text{clip}(*,0,255)$ . Это делает изображения чуть светлее и избавляет их от артефактов типа «соль» и «перец». Отсутствующее во втором ряду изображение  $\text{clip}(O5,0,255)$  визуалью совпадает с  $\text{clip}(U,0,255)$ ;
  - 2) пропорционально смасштабировать все коды матриц функцией  $\text{scale}(*,0,255)$ , сжимающей внутрибайтовые коды к центру интервала  $\Delta$  тем сильнее, чем дальше внешние коды отстоят от его границ. В результате изображение становится средне-серым и низкоконтрастным, например  $\text{scale}(O5,0,255)$ ;
  - 3) выровнять яркость изображения по всему интервалу  $\Delta$ , одновременно повысив его контраст, операцией эквализации с помощью функции *equalize*;
- из четырех алгоритмов повышения четкости границ, испытанных на конкретном затемненном и размытом изображении, лучшими после обязательной эквализации себя проявили *unicrisp* и *diacrisp*, худшим – *orthocrisp5*;

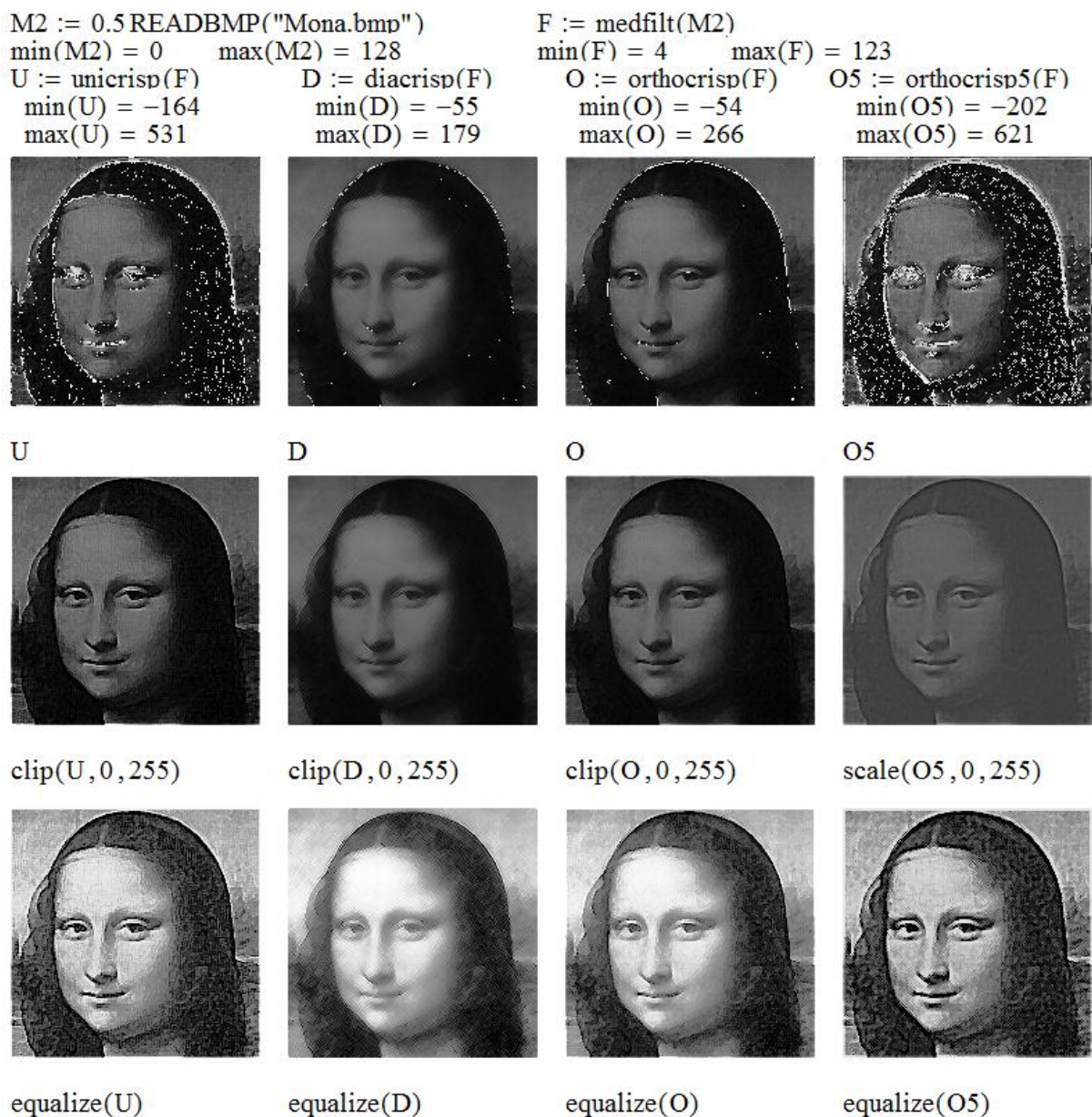


Рис. 2. Повышение четкости изображения

Fig. 2. Image enhancement

В целом следует признать, что «криповые» методы не дали значимых результатов при решении задачи повышения четкости размытых изображений. Помимо этого, они вычислительно сложнее по сравнению с методом гамма-коррекции [1, 2].

### Градиент и лапласиан

Анализ особенностей применения операций дифференцирования в задаче обнаружения границ объектов монохромных изображений начнем с рассмотрения одномерной трехбитовой дискретной функции  $f_x \in [0,7] \forall x = 0 \dots 20$  изменения яркости цепочки пикселей вдоль одной координаты  $x$ , являющейся также индексом элементов вектора яркостей  $f$ . На профиле яркости (рис. 3) выделяются относительно ровные участки с возможными включениями импульсных шумов, пологие склоны шириной более одного пиксела и скачки яркостей в обоих направлениях шириной в один пиксел.

Первая (градиент) и вторая (лапласиан) производные в точке  $f_x$

$$G_x = f_{x+1} - f_x, \quad L_x = G_x - G_{x-1} = f_{x+1} - 2f_x + f_{x-1} \quad (3)$$

как раз чувствительны к *изменениям* функции и *скорости* ее изменения и могут использоваться для поиска границ объектов – участков изображения с сильно изменяющейся яркостью. Из первого графика видно: чем круче склон в точке  $f_x$  (вплоть до ступеньки единичной ширины), тем больше модуль градиента  $|G_x|$ , а знак  $G_x$  совпадает со знаком изменения функции  $f_x$ . Для фиксации границ перепадов яркости нужны ненулевые градиенты обоих знаков.

Однако дифференцирование действует не только на полезные перепады яркости объектов, но и на малейшие шумы минимальных длительностей. Для отсеечения импульсов градиентов в точках малых шумовых флуктуаций яркости установим *порог амплитуды*  $t$ , блокирующий импульсы с  $|G_x| \leq t$ . Положительный эффект пороговой фильтрации виден на втором графике рис. 3, где импульс шума единичной высоты оказался незамеченным благодаря заданному порогу отсеечения  $t = 1$ , а все границы обоих объектов ① и ② с крутизной, большей  $t$ , обозначили себя как уровни  $w_x = 7$  условно белого цвета на черном фоне с максимальной амплитудой и шириной соответствующего перепада. Но стоит задать порог отсеечения  $t = 2$ , и граница пологого склона объекта ① с крутизной, равной 2, окажется пропущенной. В этом состоит главный недостаток градиентного метода поиска границ, требующего тщательного подбора порога отсеечения шума на основе компромисса между числом оставшихся шумов и обнаружением максимума границ объектов с пологими склонами перепадов яркости.

Для получения черных граничных линий на белом фоне достаточно сделать инверсию  $b = 7 - w$ , как показано на третьем графике (рис. 3).

```
f := stack(1, 1, 1, 3, 5, 7, 7, 7, 7, 7, 2, 2, 2, 3, 2, 2, 2, 6, 6, 6, 6)      n := last(f) = 20
x := 0..n      G_x := if(x < n, f_{x+1} - f_x, 0)      t := 1      w_x := 7(|G_x| > t)      b_x := 7 - w_x
```

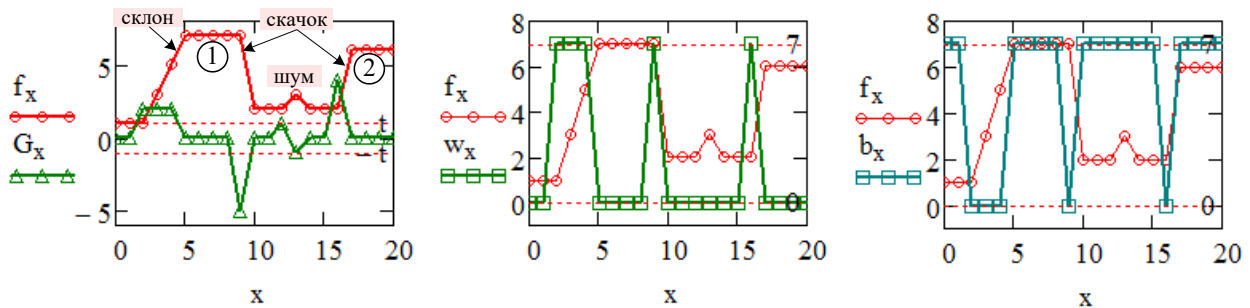


Рис. 3. Градиент одномерной функции

Fig. 3. Gradient of univariate function

Теперь составим градиент двумерной функции  $f_{xy} \in [0, 255]$  яркостей пикселей монохромного изображения, но не в виде вектора с элементами  $\partial f / \partial x = f_{x+1,y} - f_{xy}$  и  $\partial f / \partial y = f_{x,y+1} - f_{xy}$ , а как сумму изменений  $f_{xy}$  по координатам  $x$ ,  $y$  и по диагонали  $xy$ :

$$G_{xy} = (f_{x+1,y} - f_{xy}) + (f_{x,y+1} - f_{xy}) + (f_{x+1,y+1} - f_{xy}) = f_{x+1,y} + f_{x,y+1} + f_{x+1,y+1} - 3f_{xy}$$

Такое устройство скалярного градиента позволяет обнаружить не только вертикальные или горизонтальные, но и наклонные граничные линии. Его можно вычислить как свертку (1) с  $3 \times 3$ -окном  $W$  (рис. 4). В MathCAD свертка с внешне заданным окном такого размера выполняется функцией `convolve3`.

Подвергнем испытанию на поиск границ изображение  $M = \langle \text{Mona.bmp} \rangle$  (рис. 1), имеющее как резкие, так и плавные перепады яркостей пикселей. Вычисление матрицы  $G = \text{convolve3}(M, W)$ , как и ожидалось, дало разнозначные пиксельные градиенты  $G_{xy}$  в интервале  $[-408, 252]$ , превышающем  $\Delta$ , из-за чего вид изображения  $G$  нереален. Операция отсече-

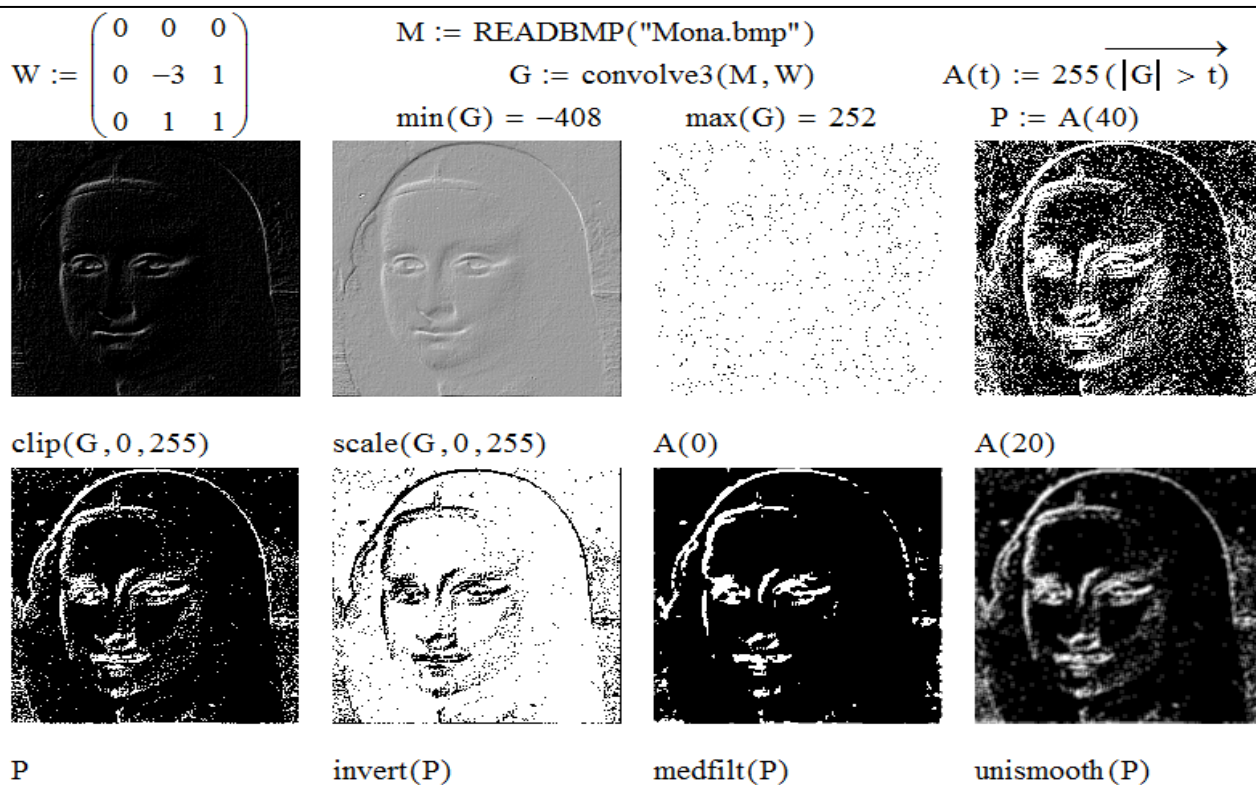


Рис. 4. Градиенты изображения

Fig. 4. Gradients of an image

ния  $\text{clip}(G, 0, 255)$  заменила отрицательные коды на 0, но светлые границы получились едва различимыми. Масштабирование  $\text{scale}(G, 0, 255)$  позволило увидеть их на светло-сером фоне, соответствующем значениям  $G_{xy}=0$ , и с чуть заметным шумом. Таким образом, разные способы приведения матрицы  $G$  в интервал  $\Delta$  не дали качественных видов границ.

Для повышения контраста изображения градиента и снижения уровня его помех выполнено попиксельное пороговое отсечение с помощью абсолютной функции  $A(t)$  и построено три пробных бинарных изображения с разными значениями порога  $t$ :

- редкие черные вкрапления на графике  $A(0)$  означают, что в изображении  $M$  очень мало точек с такой же яркостью, как у трех ближайших соседей, а, наоборот, очень много зашумленных площадей;
- в тех местах изображения  $A(20)$ , где модули градиентов  $|G_{xy}| > 20$ , появились широкие белые граничные области в окружении многочисленных импульсных шумов;
- постепенно увеличивая порог отсечения, было подобрано значение  $t=40$ , при котором на рисунке  $P=A(40)$  и его инверсии  $255-P$ , выполненной функцией  $\text{invert}$ , границы стали тонкими и более четкими, а плотность точечных помех значительно снизилась. Дальнейшее увеличение  $t$  приводит, с одной стороны, к уменьшению пиксельных шумов, но с другой – к разрывам граничных линий и превращению их в цепочки отрезков.

Последнее полезное действие с градиентным изображением – фильтрация импульсных шумов, оставшихся после пороговой обработки. Из рассмотренных в [3] методов фильтрации наиболее качественно себя проявили медианные алгоритмы  $\text{medfilt}(P)$  и идентичный ему  $\text{medfilW}(P, W)$  с окном  $W$  размера  $3 \times 3$ . Окна больших размеров, например  $5 \times 5$ , лучше удаляют импульсный шум и царапины, но еще и стирают протяженные участки граничных линий, что, конечно, неприемлемо. Не лучшими оказались и другие встроенные в MathCAD сглаживающие фильтры, например  $\text{unismooth}(P)$ .

Переходим к анализу способностей второй производной в поиске границ объектов монохромных изображений. Вновь начнем с определенного в (3) лапласиана  $L_x$  одномерной

функции  $f_x$ . Судя по его графику на рис. 5, в начале и в конце каждого склона и скачка функции создаются, в отличие от градиента, положительные либо отрицательные пиксельные импульсы. Для игнорирования малых импульсов шумовых флуктуаций яркости установим порог амплитуды  $t$ , отбирающий только импульсы с  $|L_x| > t$ . Положительные импульсы лапласиана с  $L_x > t$  дают условно белые граничные точки  $p_x = 7$  на черном фоне, а отрицательные импульсы с  $L_x < -t$  – точки  $n_x = 7$  на других краях перепадов. Пороговое отсечение по модулю  $|L_x| > t$  (как в градиенте) делать нежелательно, так как оно создаст тройные бело-черно-белые контуры вместо тонких граничных линий.

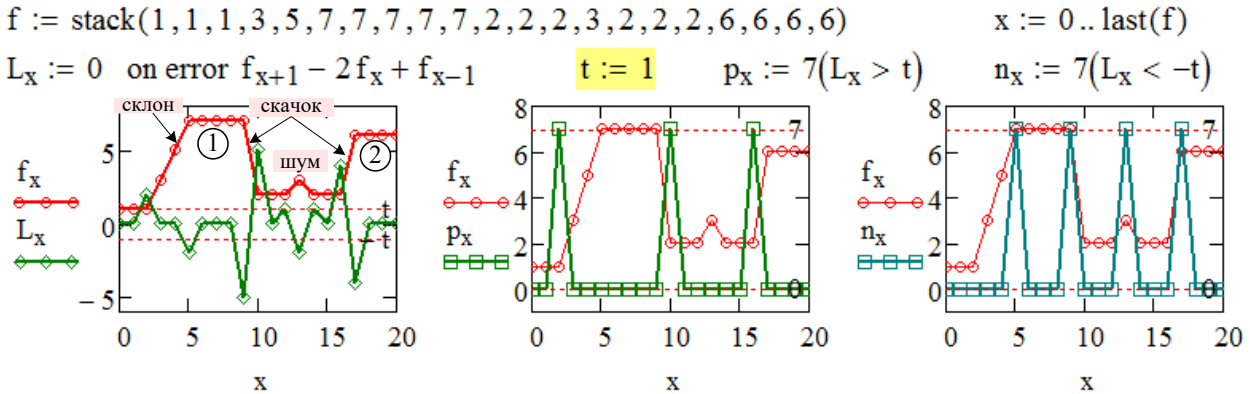


Рис. 5. Лапласиан одномерной функции

Fig. 5. Laplacian of univariate function

Теперь запишем лапласиан двумерной функции  $f_{xy} \in [0, 255]$  яркостей пикселей монохромного изображения как сумму ее вторых частных производных по координатам  $x$  и  $y$ :

$$L_{xy} = (f_{x+1,y} - 2f_{xy} + f_{x-1,y}) + (f_{x,y+1} - 2f_{xy} + f_{x,y-1}) = f_{x+1,y} + f_{x-1,y} - 4f_{xy} + f_{x,y+1} + f_{x,y-1}. \quad (4)$$

Это изотропное по горизонтали и вертикали выражение вычисляется как свертка (1) с  $3 \times 3$ -окном  $W_4$ , а после добавления в (4) диагональных лапласианов  $f_{x+1,y+1} - 2f_{xy} + f_{x-1,y-1}$  и  $f_{x+1,y-1} - 2f_{xy} + f_{x-1,y+1}$  как свертка с окном  $W_8$ . В инвертированном и нормированном виде

$$W_4 = \frac{1}{4} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}, \quad W_8 = \frac{1}{8} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}, \quad W_{24} = \frac{1}{24} \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & 24 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{bmatrix}$$

указанные окна и еще одно  $5 \times 5$ -окно  $W_{24}$  используются во встроенных функциях вычисления двумерных лапласианов  $\text{laplace4}(M)$ ,  $\text{laplace8}(M)$  и  $\text{laplace24}(M)$ , прямое действие которых на изображении «Мона.bmp», показанное на рис. 6, дало весьма далекие от желаемых сильно зашумленные результаты из-за, во-первых, двойного дифференцирования малозаметных импульсных шумов, а во-вторых, конвертирования внебайтовых кодов в ложные шумы. Попытки отсечь внебайтовые коды в матрице  $L = \text{laplace24}(M)$  функцией  $\text{clip}(L, 0, 255)$  или стянуть их в байтовый интервал функцией  $\text{scale}(L, 0, 255)$  дают слабоконтрастные изображения лапласиана, немногим лучшие, чем градиенты на рис. 4.

Очевидно, что и здесь не обойтись без средств подавления помех, желательно не размывающих обнаруженные граничные линии. Именно таким средством является амплитудный порог отсечения малых импульсов лапласиана, заданный в программе равным  $t = 10$  пикселей. Сравнивая изображения  $P$  и  $N$ , можно увидеть небольшую разницу использования положительных с  $L_{xy} > t$  и отрицательных с  $L_{xy} < -t$  импульсов лапласиана, после чего на обоих изображениях и их инверсиях все же остаются отдельные шумы, не отсеченные порогом.

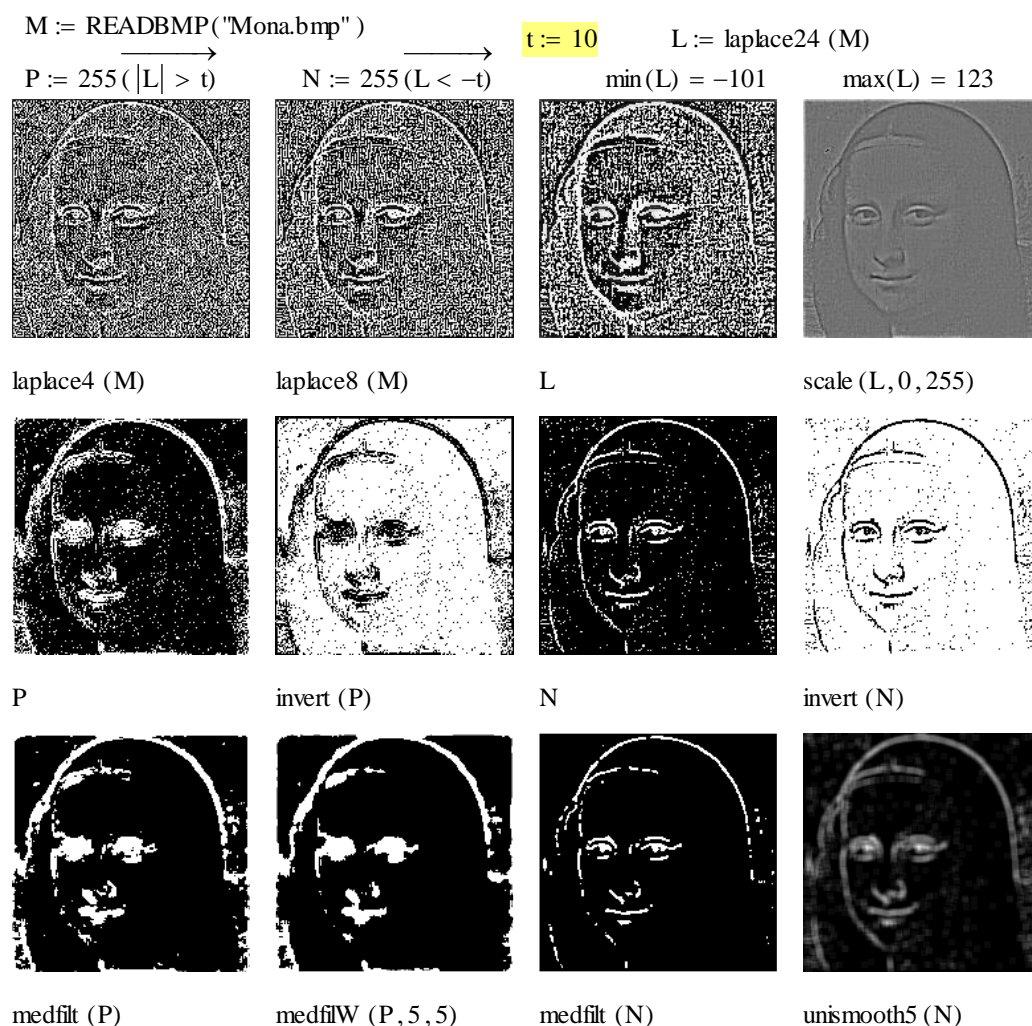


Рис. 6. Лапласианы изображения

Fig. 6. Laplacians of an image

Для окончательной подчистки этого редкого мусора применим медианную фильтрацию функцией *medfilt* с размером окна  $3 \times 3$ . Окна большего размера, например  $5 \times 5$  в функции *medfilW*, стирают импульсные шумы, но и сильнее размывают граничные линии. Неэффективны и другие сглаживающие и снижающие контраст фильтры, например *unismooth5*.

В категории «Обработка изображений» MathCAD есть ряд других встроенных функций поиска границ объектов монохромного изображения разными градиентными методами [4]. Заложенные в них алгоритмы используют расчет и анализ векторов градиента яркости пикселей. Их работа на трех изображениях *C*, *B*, *P* и *A* (рис. 1) показана на рис. 7.

Функции первой группы *difedge*, *freichen*, *prewitt*, *roberts* и *sobel* находят границы путем расчета по (1) градиентных матриц координатных свертков  $X$ ,  $Y$  с разными  $3 \times 3$ -окнами  $W_x$ ,  $W_y$  и попиксельной векторизацией модулей градиента  $\sqrt{X_{ij}^2 + Y_{ij}^2}$ . Независимо от яркости и контраста изображений, результаты поиска границ  $\text{difedge}(C) \in [0, 361]$ ,  $\text{freichen}(P) \in [0, 166]$ ,  $\text{prewitt}(A) \in [0, 252]$ ,  $\text{roberts}(B) \in [0, 266]$  и  $\text{sobel}(B) \in [0, 231]$  дают темные и низкоконтрастные изображения, яркости которых заполняют байтовый интервал  $\Delta = [0, 255]$  не полностью (слабый контраст) или выходят за его пределы (ложные шумы). Для устранения этих недостатков можно обрезать внебайтовые коды функцией *clip*, либо смасштабировать их в  $\Delta$  функцией *scale*. Однако оба эти приема дают весьма посредственный результат.

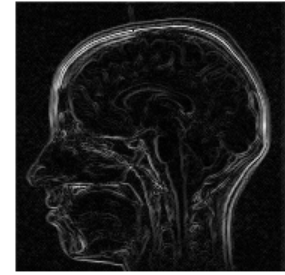


```

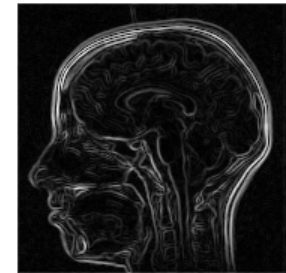
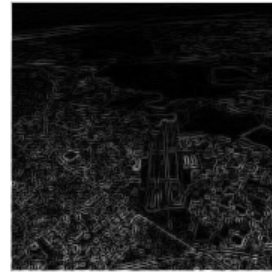
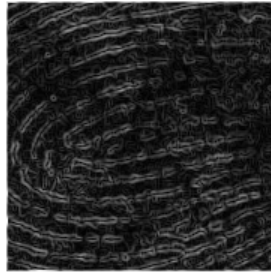
B := READBMP("Мозг.bmp") (min(B) max(B)) = (0 255)
C := READBMP("Царап.bmp") (min(C) max(C)) = (0 255)
P := READBMP("Палец.bmp") (min(P) max(P)) = (37 183)
A := READBMP("Air.bmp") (min(A) max(A)) = (30 255)

Cl(M) := clip(M,0,255) Sc(M) := scale(M,0,255)
Eq(M) := equalize(M)
Ca(M, sigma , low , high) := 255 canny(M, sigma , low , high)

```



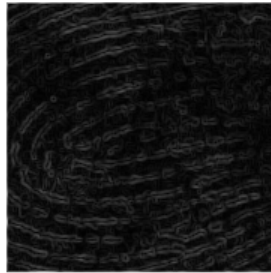
roberts(B)



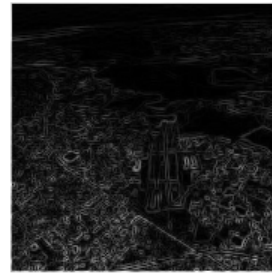
Cl(difedge(C))



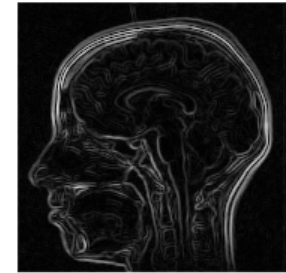
Sc(freichen(P))



prewitt(A)



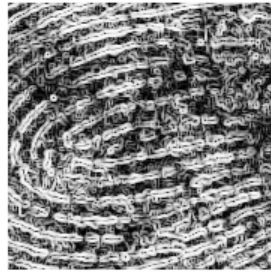
Sc(sobel(B))



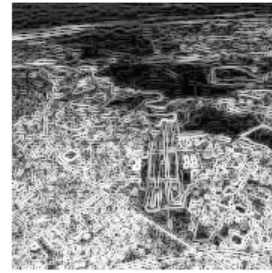
compgrad(C)



kirsch(P)



robinson3(A)



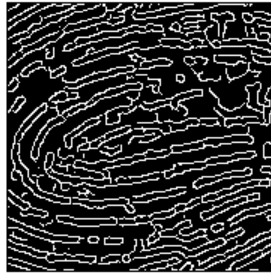
robinson5(B)



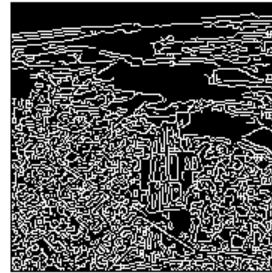
Eq(compgrad(C))



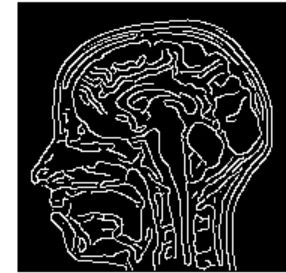
Eq(kirsch(P))



Eq(robinson3(A))



Eq(robinson5(B))



Ca(C,2,2,10)

Ca(P,2,1,4)

Ca(A,1,1,5)

Ca(B,1.5,2,6)

Рис. 7. Функции обнаружения границ

Fig. 7. Border detection functions

Функции второй группы `compgrad`, `kirsch`, `robinson3` и `robinson5` используют восемь разных матриц сверток, что, впрочем, никак не улучшает качество визуализации границ в сравнении с функциями первой группы. В третьем ряду графиков на рис. 7 применен третий способ улучшения восприятия путем выравнивания яркости изображений функцией эквализации `equalize`, которая, однако, внесла в них излишние мелкодисперсные помехи.

Функция `sanny(M, sigma, low, high)` возвращает бинарное (белые линии на черном фоне) изображение границ объектов в матрице  $M$ , найденных по алгоритму Кэнни. Он выполняет предварительную гауссовскую фильтрацию изображения с параметром  $\sigma$ . Остальные параметры гистерезисных порогов  $low$  и  $high$  используются при анализе максимальных модулей градиентных векторов  $\nabla$  в каждом исследуемом пикселе: при  $|\nabla| < low$  его интенсивность устанавливается в код фона 0, при  $|\nabla| > high$  – в код границы 1, а при  $low \leq |\nabla| \leq high$  пиксел считается граничным только если он связан хотя бы с одним из восьми граничных пикселей. Такой весьма сложный метод поиска границ позволяет экспериментально подобрать оптимальные значения параметров, находя компромисс между качеством фильтрации помех, размерами отображаемых контуров, их замкнутостью и полнотой отображения морфологической структуры изображения. Алгоритм сильно чувствителен к изменениям параметров, что и показывают примеры в нижнем ряду на рис. 7.

### Эрозия и дилатация

Один из лучших методов обнаружения границ объектов и представления их светлыми линиями на черном фоне (либо наоборот) основан на двух способах реализации двух весьма полезных операций морфологической обработки изображения  $M$ :

- сначала выполнить *эрозию* (от `erode` – разъедать, откусывать) граничных пикселей светлых объектов, закрасить их в черный цвет и вычисть суженное изображение  $E$  из  $M$ ; тогда в разности  $M - E$  останутся только пиксели, бывшие на краях светлых объектов  $M$ ;
- первой сделать *дилатацию* (от `dilate` – расширять) светлых объектов на один пиксел вглубь темного фона, затем в исходном изображении  $M$  закрасить фоновые пиксели черным цветом и вычисть из расширенного изображения  $D$  затемненное  $T$ ; тогда в разностном рисунке  $D - T$  на черном фоне останутся лишь светлые граничные контуры пиксельной толщины, окаймляющие объекты  $M$ .

Работа встроенных функций эрозии `erode4(M, b)`, `erode8(M, b)` и дилатации `dilate4(M, b)`, `dilate8(M, b)` с задаваемым граничным порогом (*threshold*) яркости  $b$ , отделяющим объекты от фона и используемым для получения матрицы  $T = \text{threshold}(M, b)$ , начинается с анализа для каждого пиксела фона  $M_{ij} < b$  яркостей соседних элементов  $M_{kl}$  (в функциях `erode4` и `dilate4` соседними являются 4 ортогональных, а в `erode8` и `dilate8` – все 8 окружающих элементов). Если существует хотя бы один соседний элемент с яркостью объекта  $M_{kl} \geq b$ , то:

- в функциях эрозии найденному светлому соседнему элементу присваивается цвет черного фона  $M_{kl} = 0$ ;
- в функциях дилатации текущему фоновому элементу с яркостью  $M_{ij} < b$  присваивается пороговое значение светлого объекта  $M_{ij} = b$ .

Испытаем описанные алгоритмы на трех монохромных изображениях с разными уровнями яркости, четкости и контраста (рис. 8). У высококонтрастного рисунка  $C = \text{«Царап.bmp»}$  с полнобайтовым диапазоном яркости  $[0, 255]$  и множеством разнотолщинных линейных и площадных помех эрозия  $EC = \text{erode8}(C, b)$ , как и ожидалось, расширила темные и сузила светлые объекты, удалив почти все белые царапины толщиной до двух пикселей (`erode4(C, b)` удаляет их чуть меньше), тогда как дилатация  $DC = \text{dilate8}(C, b)$ , наоборот, сузила темные объекты (это видно по оставшимся черным царапинам), совпав при этом с результатом функции `dilate4(C, b)`. Разностное изображение  $C - EC$  показало гораздо лучшее качество выделения границ, чем у разности  $DC - TC$ , сохранившей к тому же серые тона.

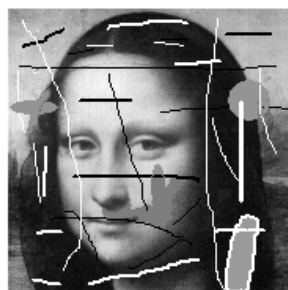
$C := \text{READBMP}(\text{"Царап.bmp"})$

$b := 128$

$EC := \text{erode8}(C, b)$

$DC := \text{dilate8}(C, b)$

$TC := \text{threshold}(C, b)$



$C$

$EC$

$C - EC$

$\text{invert}(C - EC)$



$TC$

$DC$

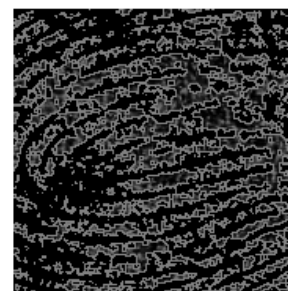
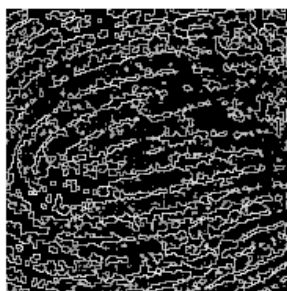
$DC - TC$

$\text{invert}(DC - TC)$

$P := \text{equalize}(\text{READBMP}(\text{"Палец.bmp"}))$

$EP := \text{erode8}(P, b)$

$DP := \text{dilate8}(P, b)$



$P$

$P - EP$

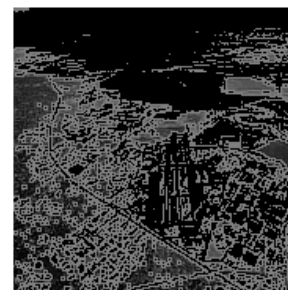
$\text{invert}(P - EP)$

$DP - \text{threshold}(P, b)$

$A := \text{equalize}(\text{READBMP}(\text{"Air.bmp"}))$

$EA := \text{erode8}(A, b)$

$DA := \text{dilate8}(A, b)$



$A$

$A - EA$

$\text{invert}(A - EA)$

$DA - \text{threshold}(A, b)$

Рис. 8. Эрозия и дилатация

Fig. 8. Erosion and dilatation

При желании увидеть черные границы объектов на белом фоне следует проинвертировать изображение  $C-EC$  функцией  $invert$ , тогда как в инвертированном разностном рисунке  $invert(DC-TC)$  все равно присутствуют серые пиксели. В связи с этим стоит из двух разностных методов отдать предпочтение использованию разности  $C-EC$ . У низкоконтрастных изображений рекомендуется сначала расширить диапазон яркостей операцией  $equalize$ , а уже потом обнаруживать границы ставших более яркими объектов. Этот прием успешно продемонстрирован на примере эквализированных рисунков  $P$  и  $A$  (рис. 1). В результате на разностных изображениях  $P-EP$  и  $A-EA$  появились контуры мельчайших светлых объектов, составляющих сюжет как темного изображения отпечатка пальца, так и засвеченной фотографии аэропорта.

Полезный способ получить граничные линии толщиной в два пиксела, не используя при этом функцию  $threshold$ , состоит в выводе разности  $D-E$  дилатированного и эродированного изображений, как показано на рис. 9. Лучшим, по мнению автора, визуальным качеством обнаружения границ обладает инвертированная разность  $invert(D-E)$ .

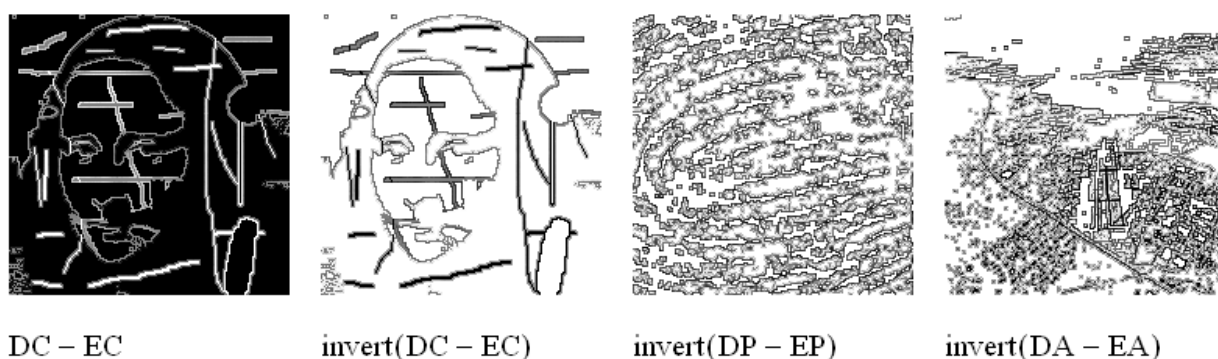


Рис. 9. Двойная толщина границ

Fig. 9. Double thickness of borders

### Выводы

Анализ качества повышения четкости растровых изображений показал, что сами по себе встроенные в MathCAD функции этого назначения особых улучшений не дали, и их рекомендуется использовать вместе с функциями обрезки, масштабирования и эквализации. Из множества функций обнаружения границ объектов наилучшие результаты показали те, что основаны на методах Кэнни, эрозии и дилатации.

### 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. Никулин, Е.А. Адаптивное управление медианной фильтрацией изображений в MathCAD [Текст] // Труды НГТУ им. Р.Е. Алексеева. 2022. № 2. С. 26-35. DOI: 10.46960/1816-210X\_2022\_2\_26.
4. Кирьянов, Д.В. Mathcad 15 / Mathcad Prime 1/0 [Текст] / Д.В. Кирьянов. – СПб.: БХВ-Петербург, 2012. – 432 с.

Дата поступления  
в редакцию: 20.12.2022

Дата принятия  
к публикации: 02.02.2023