

ЦИФРОВАЯ ОБРАБОТКА СИГНАЛОВ И ИЗОБРАЖЕНИЙ DIGITAL PROCESSING OF SIGNALS AND IMAGES

УДК 004.9

С. В. Белим, д-р физ.-мат. наук, проф., зав. каф., e-mail: sbelim@mail.ru,
С. А. Селиверстов, аспирант, e-mail: otrx@mail.ru
Омский государственный университет им. Ф. М. Достоевского, г. Омск

Использование метода анализа иерархий для выявления импульсного шума в графических объектах

Предложен алгоритм выявления поврежденных пикселей, возникающих в графических файлах в результате передачи по зашумленному каналу. Выявление поврежденных пикселей осуществляется методом анализа иерархий для принятия решения о его повреждении на основе анализа окружения. Принятие решения осуществляется на основе трех критериев. С помощью компьютерного эксперимента исследована эффективность алгоритма в зависимости от интенсивности шума.

Ключевые слова: метод анализа иерархий, импульсный шум, фильтрация изображений

Введение

Случайный шум может присутствовать на изображениях вследствие наличия помех в каналах передачи данных либо как проявление внешних воздействий на устройства получения и хранения графических данных. Случайные шумы можно разделить на несколько видов, каждый из которых проявляет себя по-разному. В данной работе мы ограничимся рассмотрением импульсного шума, проявляющегося в виде отдельных испорченных пикселей. Цвет испорченного пикселя может варьироваться в некотором интервале. В качестве частного случая импульсного шума можно рассматривать шум типа "соль и перец", в котором цвет поврежденного пикселя может принимать одно из двух значений — максимальное и минимальное.

Задача выявления импульсного шума на изображениях является одной из составляющих проблемы улучшения изображений. В большинстве решений, используемых к настоящему времени, устранение импульсного шума достигается с помощью сглаживающих фильтров. При этом речь идет не об устранении шума, а о подавлении шума в целях улучшения визуального восприятия изображения. Фильтры, подавляющие шумы, строят на основе дискретного преобразования Фурье [1, 2], вейвлет-преобразования [1, 3—5], нейронных сетей [6] и т. д. Наилучшие показатели демонстрируют нелокальные методы фильтрации [7, 8]. Все эти методы подавляют проявление шума, снижая его интенсивность, но не устраняют его полностью.

Принципиально другой подход состоит в выделении поврежденных пикселей и последующем их заполнении. Оба шага выполняются на основе ана-

лиза окружающих пикселей. Наибольших успехов на сегодняшний день удалось достичь при выявлении шума типа "соль и перец", при котором цвет поврежденного пикселя может принимать одно из двух значений, условно называемых максимальным и минимальным. Наиболее известными решениями для шума "соль и перец" являются фильтры на основе адаптивного медианного фильтра [9], масштабного параметра [10] и приближения локальной непрерывности [11]. Адаптивный медианный фильтр основан на выделении пикселей, имеющих один из цветов, присущих шуму. При использовании масштабного параметра проводится сравнение значения каждого пикселя с цветами окружающих пикселей. Решение о поврежденности принимается при превышении разностью цветов определенного порогового значения. При обнаружении поврежденных пикселей на основе предположения о локальной непрерывности считается, что наибольшая вероятность повреждения у тех пикселей, которые имеют максимальную или минимальную интенсивность.

Поврежденные пиксели, возникающие вследствие импульсного шума, во многих случаях достаточно легко различимы "на глаз". Более того, именно улучшение визуального восприятия изображения лежит в основе фильтров, подавляющих импульсные шумы. Человеческий мозг выделяет поврежденные пиксели, анализируя цвета его соседей. Следовательно, для решения задачи идентификации поврежденных пикселей могут оказаться эффективными методы Data Mining и теории принятия решений. Для применения интеллектуальных методов обработки данных необходимо формализовать

критерии, по которым человеческий глаз выделяет поврежденные пиксели.

Цель данной статьи — разработка алгоритма идентификации поврежденных пикселей, возникающих вследствие импульсного шума, с помощью метода анализа иерархий, развитого в теории принятия решений.

Постановка задачи

Пусть исходное изображение A задано в виде матрицы цветов A_{ij} . В случае цветного изображения необходимо рассматривать три матрицы согласно модели RGB. Будем считать, что диапазон значений элементов матрицы имеет вид $[0, m]$. Пусть поврежденное изображение задается матрицей цветов B_{ij} . Будем считать, что импульсный шум задан матрицей R . Каждый элемент матрицы повреждений R_{ij} с вероятностью p отличен от нуля и с вероятностью $1 - p$ — нулевой. Значение ненулевых элементов матрицы повреждений также задается случайным образом в интервале $[0, m]$. Значение матрицы поврежденного изображения задается соотношением следующим образом:

$$B_{ij} = \begin{cases} R_{ij}, & R_{ij} > 0; \\ A_{ij}, & R_{ij} = 0. \end{cases}$$

Задачей ставится определение матрицы R_{ij} по известной матрице B_{ij} .

Применение метода анализа иерархий для выявления поврежденных пикселей

Для использования метода анализа иерархий [12] необходимо четко сформулировать альтернативные решения, из которых необходимо осуществить выбор, а также критерии для анализа альтернатив. Как уже было сказано в постановке задачи, необходимо выявить пиксели, появившиеся на изображении в результате импульсного шума. Поэтому возможно только одно из двух решений, обозначаемых в дальнейшем либо Y , если данный пиксель является поврежденным, либо N , если пиксель является частью изображения. Как уже говорилось ранее, будем строить систему принятия решения на основе анализа ближайших соседей. Выделим три критерия.

K_1 — соседние по сторонам пиксели имеют тот же цвет, что и анализируемый или отличный от него. В реальной ситуации при большом многообразии цветов на данный критерий могут оказывать влияние оттенки цвета. Чтобы избежать такой ситуации, будем при анализе разбивать весь спектр цветов на m интервалов и считать, что два цвета совпадают, если они попадают в один интервал.

K_2 — соседние по углам пиксели имеют тот же цвет, что и анализируемый, или отличный от него. Как и для соседей, по сторонам будем проводить разбиение спектра на m интервалов.

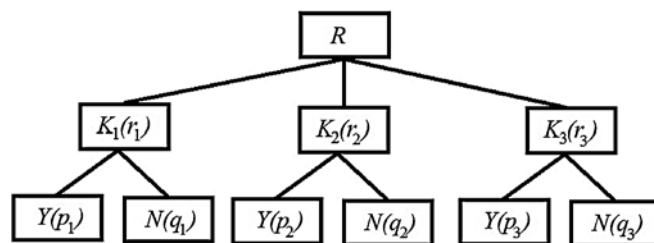


Рис. 1. Иерархия критериев для определения поврежденности пикселя

K_3 — отклонение цвета пикселя от среднего значения окружающих восьми пикселей.

Первые два критерия позволяют выявлять протяженные области изображения с равномерной заливкой. Третий критерий необходим для выявления областей с градиентной заливкой. Таким образом, получаем двухуровневое иерархическое дерево альтернатив, изображенное на рис. 1. Окончательное решение обозначено через R .

Для применения схемы анализа иерархий необходимо определить относительные веса критериев r_i ($i = 1, 2, 3$), а также веса решений в рамках одного критерия p_i и q_i ($i = 1, 2, 3$). Будем считать, что критерий K_1 важнее критерия K_2 в n раз, а критерий K_2 важнее критерия K_3 в k раз. Также будем предполагать наличие транзитивности, т. е. критерий K_1 важнее критерия K_3 в nk раз. Тогда согласованная матрица парных сравнений будет иметь следующий вид:

	K_1	K_2	K_3
K_1	1	n	kn
K_2	$1/n$	1	k
K_3	$1/(kn)$	$1/k$	1

Из данной матрицы стандартными способами [14] могут быть получены весовые коэффициенты:

$$r_1 = \frac{nk}{nk + k + 1}; r_2 = \frac{k}{nk + k + 1}; r_3 = \frac{1}{nk + k + 1}.$$

При классическом использовании метода анализа иерархий парные сравнения проводят на основе экспертных оценок. В нашем подходе вместо экспертных оценок будем использовать некоторые объективные показатели, определяемые численно. В частности, ограничения на значения n и k мы в дальнейшем определим из рассмотрения тривиальных примеров. Наиболее подходящие значения этих параметров, как и l , будут найдены из компьютерного эксперимента.

Перейдем к определению весовых коэффициентов в рамках каждого из критериев. Начнем рассмотрение с K_1 . Пусть из четырех пикселей, соприкасающихся с исследуемым, x имеют тот же цвет, тогда решение N более весомо по сравнению с Y (то есть исследуемый пиксель не испорчен) в $x/(4 - x)$ раз. Записывая матрицу парных сравнений и проводя не-

необходимые преобразования, получаем значения коэффициентов $p_1 = (4 - x)/4$, $q_1 = x/4$. Аналогично определяем коэффициенты для критерия K_2 . Пусть из четырех пикселей, соприкасающихся с данным только по вершинам, y имеют тот же цвет. Тогда весовые коэффициенты примут значения $p_2 = (4 - y)/4$, $q_2 = y/4$.

Для вычисления весовых коэффициентов по критерию K_3 предположим, что полная палитра имеет $m + 1$ цвет, т. е. максимальное значение цвета равно m . Пусть цвет исследуемого пикселя c , а среднее значение окружающих его пикселей c_0 . Для нахождения весовых коэффициентов применим следующие рассуждения. Пусть решение N более весомо, чем Y , в a раз, где значение a зависит от абсолютного значения отклонения цвета пикселя c от среднего значения окружающих пикселей c_0 ($dc = |c - c_0|$). Тогда весовые коэффициенты будут иметь следующий вид:

$$p_3 = \frac{1}{a+1}; q_3 = \frac{a}{a+1}.$$

Рассмотрим предельные случаи. В случае равенства цвета исследуемого пикселя среднему значению цветов окружающих пикселей ($dc = 0$) будем считать, что он не испорчен шумом, коэффициенты при этом будут иметь значение $p_3 = 0$, $q_3 = 1$. Если пиксель максимально отличается от окружающих ($dc = m$), то будем считать его однозначно испорченным, т. е. $p_3 = 1$, $q_3 = 0$. Следовательно, при $dc = 0$ должно быть $a \rightarrow \infty$. При значении $dc = m$ должно выполняться условие $a = 0$. Этим условиям удовлетворяет выражение

$$a = \frac{m}{dc} - 1.$$

Откуда следуют значения для весовых коэффициентов:

$$p_3 = \frac{dc}{m}, q_3 = 1 - \frac{dc}{m}.$$

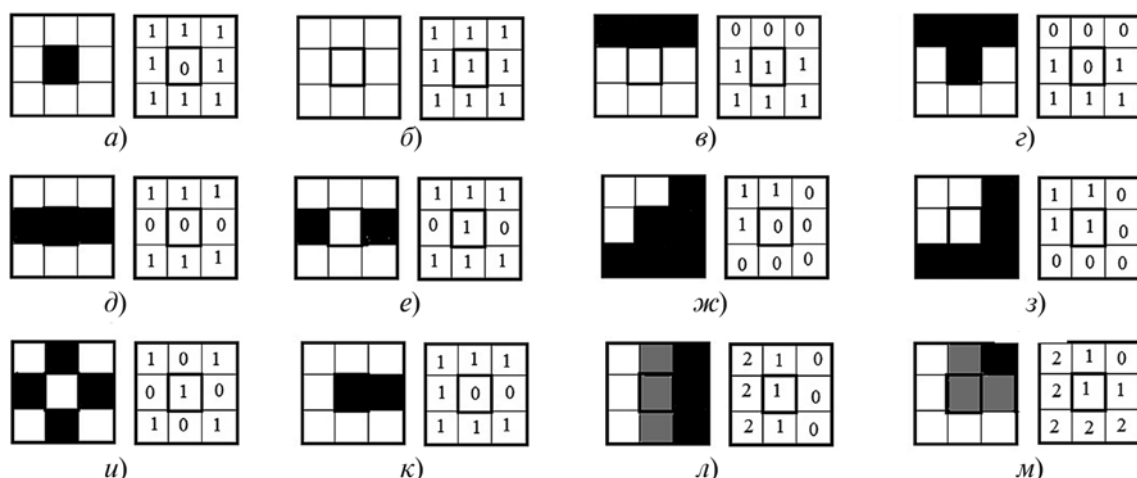


Рис. 2. Примеры расположения пикселей

Для окончательного принятия решения необходимо вычислить величины

$$P(Y) = r_1 p_1 + r_2 p_2 + r_3 p_3, P(N) = r_1 q_1 + r_2 q_2 + r_3 q_3.$$

В случае если $P(Y) > P(N)$, то принимается решение Y , т. е. пиксель является поврежденным, в противном случае, если $P(Y) \leq P(N)$, принимается решение N , т. е. пиксель является неповрежденным.

Для проверки работоспособности метода и выявления возможных ограничений на значения n и k рассмотрим несколько простейших случаев, представленных на рис. 2. Во всех случаях, представленных на этом рисунке, кроме (л) и (м), предполагается двухцветная палитра ($m = 1$). Черный цвет кодируется нулевым значением, а белый цвет — единичным. В двух последних примерах палитра трехцветная ($m = 2$) и введен промежуточный серый цвет. Во всех примерах исследуется пиксель, находящийся посередине, и принимается решение, является ли он поврежденным или нет.

Наиболее простым является пример (а), в этом случае $x = 0$, $y = 0$, $dc = 1$, соответственно, $p_1 = 1$, $q_1 = 0$, $p_2 = 1$, $q_2 = 0$, $p_3 = 1$, $q_3 = 0$, откуда $P(Y) = 1$, $P(N) = 0$, т. е. система однозначно принимает решение о поврежденности пикселя независимо от значения параметров n и k . Аналогичная ситуация наблюдается в примере (б): $x = 4$, $y = 4$, $dc = 0$, $p_1 = 0$, $q_1 = 1$, $p_2 = 0$, $q_2 = 1$, $p_3 = 0$, $q_3 = 1$, $P(Y) = 0$, $P(N) = 1$, т. е. при любых значениях n и k система принимает решение о неповрежденности пикселя. Из первых двух примеров можно сделать вывод, что предложенный метод однозначно находит единственный поврежденный пиксель, окруженный одноцветной заливкой.

В случае (в): $x = 3$, $y = 2$, $dc = 3/8$, $p_1 = 1/4$, $q_1 = 3/4$, $p_2 = 1/2$, $q_2 = 1/2$, $p_3 = 3/8$, $q_3 = 5/8$,

$$P(Y) = \frac{1}{4}r_1 + \frac{1}{2}r_2 + \frac{3}{8}r_3 = \frac{2nk+4k+3}{8(nk+k+1)}, P(N) = \frac{3}{4}r_1 + \frac{1}{2}r_2 + \frac{5}{8}r_3 = \frac{6nk+4k+5}{8(nk+k+1)}.$$

При любых положительных n и k будет выполняться неравенство $P(Y) < P(N)$, т. е. пиксель будет определен как неповрежденный. В случае (з): $x = 1$, $y = 2$, $dc = 5/8$, $p_1 = 3/4$, $q_1 = 1/4$, $p_2 = 1/2$, $q_2 = 1/2$, $p_3 = 5/8$, $q_3 = 3/8$, $P(N) = \frac{1}{4}r_1 + \frac{1}{2}r_2 + \frac{3}{8}r_3 = \frac{2nk+4k+3}{8(nk+k+1)}$, $P(Y) = \frac{3}{4}r_1 + \frac{1}{2}r_2 + \frac{5}{8}r_3 = \frac{6nk+4k+5}{8(nk+k+1)}$.

Следовательно, $P(Y) > P(N)$ и пиксель будет распознан как поврежденный. Из двух последних примеров видно, что предложенный метод позволяет выявлять поврежденные пиксели, расположенные вблизи горизонтальной границы. В силу симметричности предложенных критериев данный вывод останется верным и для вертикальной границы, а также при инвертировании цветов.

В случае (д): $x = 2$, $y = 0$, $dc = 3/4$, $p_1 = 1/2$, $q_1 = 1/2$, $p_2 = 1$, $q_2 = 0$, $p_3 = 3/4$, $q_3 = 1/4$, $P(Y) = 0,5r_1 + r_2 + 0,75r_3$, $P(N) = 0,5r_1 + 0,25r_3$. Так как все весовые коэффициенты положительные, то $P(Y) > P(N)$, следовательно, линии толщиной в один пиксель будут восприниматься как испорченные. Этот же вывод подтверждается рассмотрением примера (е), для которого $P(Y) < P(N)$.

Для случая (ж): $x = 2$, $y = 3$, $dc = 3/8$, $p_1 = 1/2$, $q_1 = 1/2$, $p_2 = 1/4$, $q_2 = 3/4$, $p_3 = 3/8$, $q_3 = 5/4$, $P(Y) = 0,5r_1 + 0,25r_2 + 0,125r_3$, $P(N) = 0,5r_1 + 0,75r_2 + 0,625r_3$. Очевидно, выполняется неравенство $P(Y) < P(N)$, т. е. метод позволяет правильно обрабатывать границы, расположенные под углом. В близком к нему случае (з) будет выполняться неравенство $P(Y) > P(N)$, значит, пиксели, расположенные в углах, будут восприниматься как поврежденные.

Интерес представляет случай (и). Если положим, что $n > 1$, т. е. горизонтальные структуры важнее диагональных, тогда $P(Y) > P(N)$ и диагностируется поврежденность пикселя. В обратном случае $n < 1$ будет выполняться неравенство $P(Y) < P(N)$ и пиксель будет считаться неповрежденным. В случае (к), как и следовало ожидать, пиксель будет определяться как поврежденный при любых значениях n и k .

Рассмотрим случаи присутствия градиента цвета. В случае (л) пиксель будет считаться неиспорченным при условии $k < 1$. Рассмотрение случая (м) и требование определения пикселя как неиспорченного приводит к более сильному условию $k < 1/2$.

Таким образом, из рассмотренных примеров следует, что для правильной обработки основных структур на изображении необходимо выполнение условий $n > 1$ и $k < 1/2$.

Алгоритм выявления поврежденных пикселей

Запишем формально алгоритм, реализующий предложенный метод. Осуществляем последовательный проход по всем пикселям изображения. Для каждого пикселя выполняем последовательность шагов.

Шаг 1. Определяем цвета ближайших соседей.

Шаг 2. Вычисляем сравнительные характеристики x , y , dc и на их основе весовые коэффициенты p_i , q_i ($i = 1, 2, 3$).

Шаг 3. Вычисляем значения вероятностей $P(Y)$ и $P(N)$.

Шаг 4. Если $P(Y) > P(N)$, то заносим координаты пикселя в карту поврежденных пикселей.

Шаг 5. Переходим к следующему пикселю.

На выходе алгоритма будет получен список поврежденных пикселей. Так как в алгоритме осуществляется один проход по всем пикселям и для каждого пикселя выполняется фиксированное число шагов, то трудоемкость алгоритма будет линейной. Также следует отметить локализацию данных, необходимых для принятия решения, в малой области вокруг исследуемого пикселя, что позволяет легко выполнять распараллеливание алгоритма простым разбиением изображения на области.

Компьютерный эксперимент и результаты

Для исследования эффективности предложенного метода был проведен компьютерный эксперимент по выявлению поврежденных пикселей. Исследования проводили на трех типах изображений: однородной заливке, искусственном изображении геометрических фигур и широко используемом изображении "Lena". Все изображения имели размер 256×256 пикселей, глубина цвета составляла 256 цветов. С помощью генератора псевдослучайных последовательностей с равномерным распределением выбирали координаты поврежденного пикселя и его новый цвет.

Исследования проводили при различных значениях относительного числа поврежденных пикселей p от 10 до 70 %. Разбиение спектра цветов на интервалы задавалось числом интервалов m . В нашем компьютерном эксперименте m варьировалось от 1 до 128. Работа алгоритма оценивалась по двум параметрам: *eff* — эффективность, т. е. число правильно найденных поврежденных пикселей к общему числу сгенерированных поврежденных пикселей; *err* — процент ложных срабатываний как отношение ошибочно определенных пикселей к общему числу пикселей, найденных алгоритмом.

На рис. 3 (см. четвертую сторону обложки) представлен равномерно закрашенный фон с импульсным шумом интенсивности 10 %, а также карта поврежденных пикселей для $m = 8$ и $m = 24$. Как показали численные оценки, в первом случае было обнаружено 85 % поврежденных пикселей, а во вто-

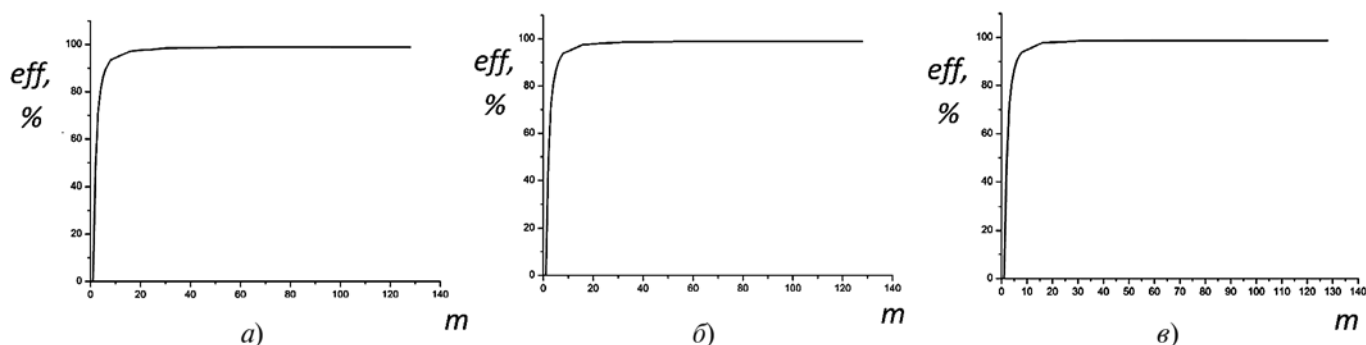


Рис. 6. Зависимость процента верно обнаруженных пикселей от глубины палитры m :
 $a - p = 10$; $b - p = 20$; $v - p = 30$

ром случае 94 % поврежденных пикселей. Ложные срабатывания составляли в среднем 0,2 % в обоих случаях. Были обнаружены не все поврежденные пиксели в связи с тем, что цвет их задавался случайно и некоторые из них попали в тот же цветовой интервал, что и основной фон. Наличие ложных срабатываний обусловлено попаданием поврежденных пикселей в соседние точки изображения.

Более интересные результаты были получены для искусственного изображения геометрических фигур с четкими границами. Результаты представлены на рис. 4 (см. четвертую сторону обложки) для случая $p = 10$ %. Как видим, при $m = 8$ достаточно хорошо определяются одиночные поврежденные пиксели. Численная оценка показывает, что верно определены 87 % пикселей, но при этом ложные срабатывания составляют 9 %. При $m = 24$ верно определяются 93 % пикселей, ложные срабатывания составляют также 9 %. Большой процент ложных срабатываний связан с тем, что алгоритм, кроме поврежденных пикселей, также находит резкие границы на изображении (рис. 4, $a, б$, см. четвертую сторону обложки).

Результаты работы на реальном изображении "Lena" представлены на рис. 5 (см. четвертую сторону обложки). При $m = 8$ верно определено 85 % поврежденных пикселей, ложные срабатывания 55 %. При $m = 24$ верно определено 94 % поврежденных пикселей, ложные срабатывания 77 %. Как видим

на реальном изображении процент ложных срабатываний достаточно высок. Это связано с тем, что в выбранном изображении "Lena" много мелких деталей, которые воспринимаются алгоритмом как поврежденные пиксели. В частности, при увеличении числа интервалов деления спектра цветов в качестве испорченных алгоритм воспринимает все резкие границы, что хорошо видно на рис. 5, v .

Далее эксперименты проводили на основе изображения "Lena" как широко известного и обладающего всеми характерными свойствами реальных графических объектов. Как было видно из рис. 5, при увеличении числа цветов в палитре изменяется как эффективность обнаружения поврежденных пикселей, так и процент ложных срабатываний. На рис. 6 представлены зависимости процента верно опознанных поврежденных пикселей от числа цветов в палитре m при различных уровнях загрязнения p . Как видно из этого рисунка, во всех случаях наблюдается схожая картина быстрого роста эффективности обнаружения при $m < 20$ с дальнейшим переходом к очень медленному росту. Насыщения в явном виде не наступает. Кривая медленно приближается к значению 100 %.

На рис. 7 представлены зависимости процента ложных срабатываний от числа цветов в палитре m при различных уровнях загрязнения p . Как видно из рисунка, во всех случаях присутствует явный минимум около значения $m = 5$, дальнейшее увели-

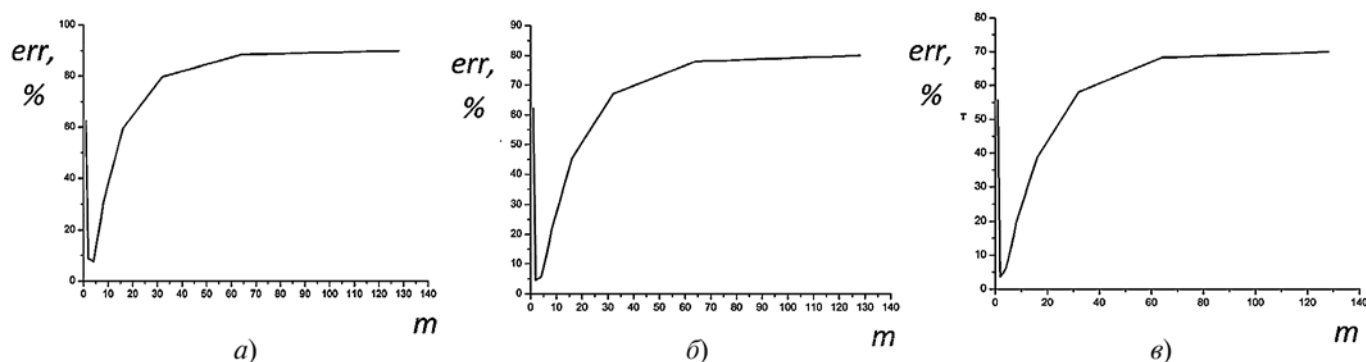


Рис. 7. Зависимость процента ложных срабатываний от глубины палитры m :
 $a - p = 10$; $b - p = 20$; $v - p = 30$

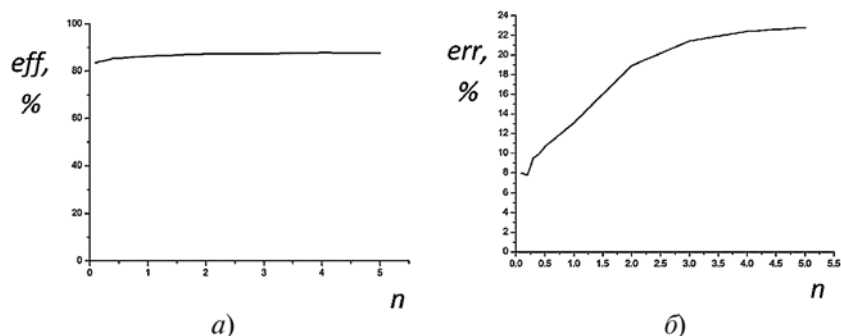


Рис. 8. Зависимость эффективности обнаружения eff (а) и процента ложных срабатываний err (б) от весового коэффициента n при $m = 5$ и $k = 1$

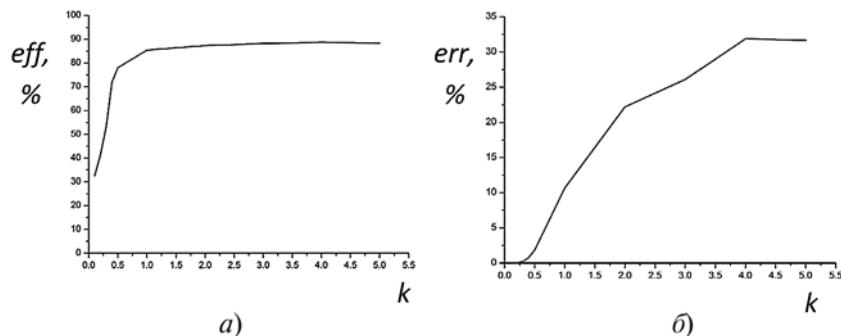


Рис. 9. Зависимость эффективности обнаружения eff (а) и процента ложных срабатываний err (б) от весового коэффициента k при $m = 5$ и $n = 0,5$

чение числа цветов m ведет к росту процента ложных срабатываний. Данное поведение графика легко объяснимо. Если используется разбиение на малое число интервалов, то предложенный метод не очень чувствителен к разнице цветов, вследствие чего обнаруживается мало пикселей как верных, так и неверных. При этом метод ошибочно находит все резкие границы. С ростом m уменьшается размер цветовых интервалов, что позволяет более точно определять поврежденные пиксели, реагируя при этом ошибочно только на резкие границы. Если же интервалы цветов сделать еще меньше, то верных поврежденных пикселей будет обнаруживаться незначительно больше, тогда как алгоритм будет ошибочно срабатывать на все более гладких границах областей изображения.

Ранее во всех случаях мы считали, что весовые коэффициенты n и k , определяющие относительную важность различных критериев, равны единице. Это приближение было выбрано, основываясь на простых случаях, рассмотренных в предыдущем разделе. Тем не менее этот вопрос требует дополнительного изучения на реальных изображениях. Был проведен компьютерный эксперимент при $m = 5$ для различных значений n и k . На рис. 8 представлена зависимость эффективности восстановления от параметра n при $k = 1$. Как хорошо видно из рисунка, эффективность восстановления практически не зависит от весового коэффициента n , тогда как процент ложных срабатываний монотонно возрастает.

Далее был проведен эксперимент по исследованию зависимости эффективности восстановления и процента ложных срабатываний от весового коэффициента k при $m = 5$ и $n = 0,5$ (рис. 9). Как видно из графиков, эффективность определения поврежденных пикселей растет на интервале от 0 до 1, а далее остается практически неизменной. Для процента ложных срабатываний наблюдается монотонный рост на всем исследованном интервале.

При реализации прикладной системы идентификации поврежденных пикселей необходимо выбирать оптимальные параметры системы m , n и k . Однако в данном случае мы имеем дело с задачей многофакторной оптимизации. Необходимо, чтобы одновременно была наибольшая эффективность определения поврежденных пикселей и минимальный процент ложных срабатываний. Для решения этой задачи может быть использован метод взвешенной суммы [13] при построении целевой функции:

$$F(m, n, k) = a_1 eff(m, n, k) + \frac{a_2}{err(m, n, k)} \rightarrow \min.$$

Положительные коэффициенты a_1 и a_2 выбираются из соображений значимости того или иного критерия и удовлетворяют условию $a_1 + a_2 = 1$.

Обсуждение результатов и выводы

Предложенный в данной статье алгоритм выявления импульсного шума на изображениях достаточно эффективен как на искусственных, так и на естественных изображениях. Также следует отметить предложенное нами нетрадиционное использование метода анализа иерархий для принятия решения. В обычном подходе [12] весовые коэффициенты по каждому из критериев определяются на основе экспертных оценок и носят субъективный характер. В предложенном подходе экспертные оценки вычисляются автоматически, исходя из локальных свойств исследуемой части изображения, и носят объективный характер. Повышение эффективности предложенного метода может быть достигнуто путем введения дополнительных критериев, а также с помощью анализа пикселей не только являющихся ближайшими соседями, но и следующих за ними.

Проведем сравнение полученных нами результатов с результатами, полученными в других работах. Следует отметить, что в подавляющем большинстве

работ ставится задача сглаживания импульсного шума для улучшения восприятия изображения, а не явного обнаружения испорченных пикселей. Существующие алгоритмы обнаружения делят на два больших класса: обнаруживающие шум типа "соль и перец" (Salt & Pepper Noise SPN) и обнаруживающие импульсный шум случайного значения (Random-Value Impulse Noise — RVIN). Алгоритмы для обнаружения SPN имеют высокую эффективность в силу того, что заранее известны цвета поврежденных пикселей. Однако и эти алгоритмы характеризуются наличием необнаруженных пикселей. Так, например, в работе [14] проведено тестирование четырех различных алгоритмов обнаружения импульсного шума типа "соль и перец", для которых эффективность обнаружения поврежденных пикселей составляет от 71 до 96 % поврежденных пикселей при импульсном шуме с плотностью $p = 30$ %. Задача выявления импульсного шума случайного значения является значительно более трудной по сравнению с выявлением SPN, однако наш алгоритм показывает сопоставимые по эффективности результаты. Методы выявления поврежденных пикселей для импульсного шума случайного значения в большинстве случаев строят на основе алгоритма SD-ROM [15–18]. Данный алгоритм был предложен в работе [15] и основан на анализе пикселей, попадающих в скользящее окно размером 3×3 , содержащее в центре исследуемый пиксель. Решение о поврежденности пикселя принимается на основе пороговой схемы. Причем значения четырех параметров, играющих роль пороговых значений, выбирает пользователь произвольно из некоторого интервала, и они зависят от самого изображения. При правильном выборе параметров алгоритм способен давать результаты, сопоставимые с алгоритмами для шума типа "соль и перец". При ошибочном выборе параметров эффективность резко падает. Предложенный алгоритм дает стабильные значения эффективности поиска поврежденных пикселей для различных изображений, не уступает аналогичным разработкам и не требует ручной подгонки параметров.

Следует отметить, что предлагаемый подход является только первым этапом при восстановлении изображений, поврежденных импульсным шумом. Поэтому его результаты надо воспринимать как предобработку графических данных. Как хорошо известно, повреждения изображений импульсным шумом не являются критическими вплоть до значения p , равного около 75 % [19]. Предложенный алгоритм при таком уровне шума в качестве поврежденных пикселей начинает определять также и границы областей. Однако определяемые границы имеют толщину в один пиксель и при дальнейшем восстановлении слабо сказываются на общем виде восстановленного изображения. Данные погрешности алгоритма могут приводить к небольшому размытию границ, что является характерной осо-

бенностью всех фильтров, устраняющих импульсный шум.

Работа проведена при финансовой поддержке Министерства образования и науки РФ в рамках государственного задания вузам в части проведения научно-исследовательских работ на 2014–2016 гг., проект № 2314.

Список литературы

1. Astola J., Kuosmanen P. Fundamentals of Nonlinear Digital Filtering. Boca Raton, FL: CRC Press, 1997.
2. Kijewski M., Judy P. The noise power spectrum of ct images // Phys. Med. Biol. 1987. V. 32 (5). P. 565–575.
3. Peng-Lang Sui. Image Denoising Algorithm via Doubly Local Wiener Filtering With Directional Windows in Wavelet Domain // IEEE Transactions on Signal Processing Letters. October 2005. V. 12, N. 12. P. 681–684.
4. Petrović N., Crnojević V. Impulse Noise Detection Based on Robust Statistics and Genetic Programming // Lecture Notes in Computer Science. 2005. V. 3708. P. 643–649.
5. Miller M., Kingsbury N. Image Denoising Using Derotated Complex Wavelet Coefficients // IEEE Transactions on Image Processing. September 2008. V. 17, N. 9. P. 1500–1511.
6. Kong H., Guan L. Detection and removal of impulse noise by a neural network guided adaptive median filter // Neural Networks. Proceedings. IEEE International Conference on 1995. V. 2. P. 845–849.
7. Buades A., Morel J. A non-local algorithm for image denoising // IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2005. P. 60–65.
8. Dabov K., Foi A., Katkovnik V., Egiazarian K. Image denoising by sparse 3d transform-domain collaborative filtering // IEEE Trans. Image Process. 2007. V. 16, N. 8. P. 2080–2095.
9. Chan R., Ho C., Nikolova M. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization // IEEE Trans. Image Process. 2005. V. 14, N. 10. P. 1479–1485.
10. Kam H. S., Tan W. H. Noise detection fuzzy (NDF) filter for removing salt and pepper noise // Lecture Notes Comput. Sci. 2009. V. 5857. P. 479–486. (Intern. visual informatics conf. 2009 (IVIC'09)).
11. Najeer A. J., Rajamahi V. Design of hybrid filter for denoising images using fuzzy network and edge detecting // Amer. J. Sci. Res. 2009. Iss. 3. P. 5–14.
12. Saaty T. L. Relative Measurement and its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors — The Analytic Hierarchy / Network Process. // Review of the Royal Spanish Academy of Sciences, Series A, Mathematics. 2008. V. 102 (2). P. 251–318.
13. Jong-Hyun Ryu, Sujin Kim, Hong Wan. Pareto front approximation with adaptive sum method in multiobjective simulation optimization // Proc. of the 2009 Winter Simulation Conference (WSC). Austin. 2009. P. 623–633.
14. Нрок Хоанг Фан, Спицын В. Г. Анализ алгоритмов обнаружения импульсного шума на цифровых изображениях // Известия Томского политехнического университета. 2011. Т. 318, № 5. С. 70–73.
15. Abreu E., Lightstone M., Mitra S. K., Arakawa S. K. A new efficient approach for the removal of impulse noise from highly corrupted images. // IEEE Transactions on Image Processing, IEEE Transactions on. 1996. V. 5. P. 1012–1025.
16. Garnett R., Huegerich T., Chui C., He W. A Universal Noise Removal Algorithm with an Impulse Detector // IEEE Trans Image Process. 2005. N. 14 (11). P. 1747–54.
17. Красовский Г. Я., Усс М. Л. Фильтрация изображений, искаженных импульсными помехами точечного и строчного типа, на основе систем итерированных функций // Радиоэлектронні і комп'ютерні системи. 2003. В. 2. С. 47–55.
18. Сорокин С. В., Щербаков М. А. Реализация SD-ROM фильтра на основе концепции нечеткой логики // Известия высших учебных заведений. Поволжский регион. 2007. № 3. С. 56–65.
19. Hutchinson J. Culture, Communication, and an Information age Madonna // IEEE Professional Communication Society Newsletter. May/June 2001. V. 45, N. 3. P. 1, 5.

Hierarchy Analysis Method as a Way to Detect Impulse Noise on Images

This article presents an algorithm for identifying broken pixels, that can appear in graphic files consequently to their transfer through noisy channel. The detection of broken pixels is carried through the sequential check of image file. The method of hierarchical analysis is used for each pixel in order to arrive at a decision about its impairment through the analysis of its environment. The decision is based on 3 kinds of criteria. A case of impulse noise — randomly situated pixels of different colors, is analyzed. The algorithm efficiency depending on the noise intensity is also investigated through a computed experiment. In the present article it is proved that with the help of parameter design the efficiency of detecting broken pixels can reach 96 %. The computer experiment helped to reveal the dependence of the proposed method efficiency against various parameters. The article also presents the investigation of false response percentage depending on selected parameters. It is shown that the efficiency can amount up to 90 % whilst the 8 % of positive response.

Keywords: hierarchy analysis method, impulse noise, image filtering

References

1. Astola J., Kuosmanen P. *Fundamentals of Nonlinear Digital Filtering*. Boca Raton, FL: CRC Press, 1997.
2. Kijewski M., Judy P. The noise power spectrum of ct images. *Phys. Med. Biol.* 1987. V. 32 (5). P. 565—575.
3. Peng-Lang Sui. Image Denoising A lgorithm via Doubly Local Wiener Filtering With Directional Windows in Wavelet Domain. *IEEE Transactions on Signal Processing Letters*. October 2005. V. 12, N. 12. P. 681—684.
4. Petrović N., Crnojević V. Impulse Noise Detection Based on Robust Statistics and Genetic Programming. *Lecture Notes in Computer Science*. 2005. V. 3708. P. 643—649.
5. Miller M., Kingsbury N. Image Denoising Using Derotated Complex Wavelet Coefficients. *IEEE Transactions on Image Processing*. September 2008. V. 17, N. 9. P. 1500—1511.
6. Kong H., Guan L. Detection and removal of impulse noise by a neural network guided adaptive median filter. *Neural Networks. Proceedings. IEEE International Conference on*. 1995. V. 2. P. 845—849.
7. Buades A., Morel J. A non-local algorithm for image denoising *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2. 2005. C. 60—65.
8. Dabov K., Foi A., Katkovnik V., Egiazarian K. Image denoising by sparse 3d transform-domain collaborative filtering *IEEE Trans. Image Process.* 2007. V. 16 (8). P. 2080—2095.
9. Chan R., Ho C., Nikolova M. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Trans. Image Proc.* 2005. V. 14, N. 10. P. 1479—1485.
10. Kam H. S., Tan W. H. Noise detection fuzzy (NDF) filter for removing salt and pepper noise. *Lecture Notes Comgut. Sci.* 2009. V. 5857. P. 479—486. (*Intern. visual informatics conf. 2009 (IVIC'09)*).
11. Najeer A. J., Rajamani V. Design of hybrid filfeT for denoising images using fuzzy network and edge detecting. *Amer. J. Sci. Res. Iss.* 3. 2009. P. 5—14.
12. Saaty T. L. Relative Measurement and its Generalization in Decision Making: Why Pairwise Comparisons are Central in Mathematics for the Measurement of Intangible Factors — The Analytic Hierarchy / Network Process. *Review of the Royal Spanish Academy of Sciences. Series A, Mathematics*. 2008. V. 102 (2). P. 251—318.
13. Jong-hyun Ryu, Sujin Kim, Hong Wan. Pareto front approximation with adaptive sum method in multiobjective simulation optimization. *Proc. of the 2009 Winter Simulation Conference (WSC)*. Austin. 2009. P. 623—633.
14. Ngok Hoang Fan, Spitsyn V. G. Analiz algoritmov obnaruzheniya impulsnogo shuma na tsifrovyyh izobrazheniyah. *Izvestiya Tomskogo politehnicheskogo universiteta*. 2011, V. 318, N. 5. P. 70—73.
15. Abreu E., Lightstone M., Mitra S. K., Arakawa S. K. A new efficient approach for the removal of impulse noise from highly corrupted images. *IEEE Transactions on Image Processing, IEEE Transactions on*. 1996, V. 5. P. 1012—1025.
16. Garnett R., Huegerich T., Chui C., He W. A Universal Noise Removal Algorithm with an Impulse Detector. *IEEE Trans Image Process.* 2005. N. 14 (11). P. 747—54.
17. Krasovskiy G. Ya., Uss M. L. Filtratsiya izobrazheniy, iskazhenyih impulsnyimi pomehami tochechnogo i strochnogo tipa, na osnove sistem iterirovanyih funktsiy. *Radioelektronni i komp'yuterni sistemi*. 2003. N. 2. P. 47—55.
18. Sorokin S. V., Scherbakov M. A. Realizatsiya SD-ROM filtra na osnove kontseptsii nechetkoy logiki. *Izvestiya vysshih uchebnyih zavedeniy. Povolzhskiy region*. 2007. N 3. P. 56—65.
19. Hutchinson J. Culture, Communication, and an Information age Madonna. *IEEE Professional Communication Society Newsletter*. May/June 2001. V. 45, N. 3. P. 1, 5.