Наука и Образование
МГТУ им. Н.Э. Баумана
Сетевое научное издание
ISSN 1994-0408

Наука и Образование. МГТУ им. Н.Э. Баумана. Электрон. журн. 2014. № 12. С. 716–737.

DOI: **10.7463/1214.0744983**

Представлена в редакцию: 13.07.2014
Исправлена: 18.11.2014

© МГТУ им. Н.Э. Баумана

УДК 51(083)

Алгоритм поиска поврежденных пикселей и удаления импульсного шума на изображениях с использованием метода ассоциативных правил

Белим С. В.[1,*], *sbelim@mail.ru
Майоров-Зильбернагель А. О.[1]

[1]Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

В статье предложен алгоритм, позволяющий на основе анализа изображения с помощью метода построения ассоциативных правил выявлять пиксели, поврежденные импульсным шумом. Предложенный алгоритм состоит из двух этапов. На первом этапе происходит построение ассоциативных правил. Первый этап имеет линейную трудоемкость, однако требует большого объема памяти. На втором этапе осуществляется повторный проход по изображению и выявление испорченных пикселей. Также предложенный подход позволяет восстанавливать поврежденные изображения, на основе ассоциативных правил выявленных на первом этапе. Проведен компьютерный эксперимент по определению эффективности предложенного метода. Выявлена зависимость эффективности работы алгоритма от различных параметров. Эффективность поиска поврежденных пикселей составляет не менее 76,9% на естественных изображениях. Для изображений с большим количеством мелких деталей, наблюдается достаточно высокий процент ложных срабатываний.

Ключевые слова: восстановление данных, зашумленные изображения, ассоциативные правила, импульсный шум

Введение

Проблема выявления имульсного шума на изображениях является одной из наиболее важных при передачи данных по каналам с помехами [1]. Импульсный шум проявляет себя как отдельные случайно расположенные на изображении пиксели, имеющие случайный цвет [2,3]. Наиболее распространенным подходом к решению этой задачи на сегодняшний день является нивелирование последствий импульсного шума с помощью сглаживающих фильтров. Данный подход визуально устраняет поврежденные пиксели, но одновременно ведет к размытию исходного изображения. Принципиально иной подход состоит в применении систем искусственного интеллекта для выявления поврежденных

пикселей и построение маски повреждений. Далее поврежденные пиксели рассматриваются как утерянные данные. Таким образом, задача сводится к восстановлению таблицы с пропусками.

На сегодняшний день разработано достаточно много сглаживающих фильтров, позволяющих визуально убирать импульсный шум. К таким фильтрам следует, прежде всего, отнести фильтр Винера и медианный фильтр [4]. Наилучших показателей позволяют добиться нелокальные методы фильтрации [5-7]. Фильтры, подавляющие шумы, визуально устраняя шумы, вносят также изменения в неповрежденные части изображения, размывая резкие границы. В качестве одного из способов устранения этого побочного явления используется дополнительная фильтрация с повышением резкости границы, что еще сильнее искажает исходное изображение.

Интеллектуальный подход к анализу зашумленных изображений состоит в применении алгоритма выделения и заполнения поврежденных пикселей на основе методов Data Mining. Как и в случае использования сглаживающих фильтров, каждый пиксель обрабатывается, исходя из цвета окружающих пикселей. Однако результат обработки основывается не на применении неизменных алгебраических преобразований, а на методах принятия решений. Большинство предложенных к сегодняшнему дню алгоритмов ориентированы на выявление шума типа "соль и перец". В этом случае цвет поврежденного пикселя может принимать одно из двух значений - максимальное и минимальное в используемой палитре. В этом случае хорошие показатели демонстрируют медианный фильтр [8], фильтр на основе масштабного параметра [9] и фильтр на основе приближения локальной непрерывности [10].

Для заполнения поврежденных пикселей могут быть использованы различные алгоритмы восстановления таблиц с пропусками. Наиболее простым и распространенным является заполнение по среднему значению окружающих ячеек [11]. Среднее по соседям дает приемлемые результаты на участках изображения без резких границ. Также возможен подход, использующий построение регрессионной модели [12]. Использование интерполяции сплайнами [13] основывается на предположении о наличии непрерывной функции, совпадающей с изображением в точках с целочисленными координатами. Хорошие результаты могут быть получены на основе факторного анализа [14], однако данный подход обладает высокой трудоемкостью. Также возможно применение кластерного анализа [15], позволяющего выявлять ситуации, когда восстановление данных невозможно. Алгоритмы восстановления таблиц с пропусками семейства ZET [16,17] основаны на

гипотезе об избыточности табличных данных. Также для восстановления пропущенных данных активно применяются искусственные нейронные сети [18,19].

В используемых в настоящее время алгоритмах обнаружения испорченных пикселей можно выделить на два больших класса: предназначенные для обнаружения шума типа «Соль и перец» (Salt & Pepper Noise SPN) и предназначенные для обнаружения импульсного шума случайного значения (Random-Value Impulse Noise - RVIN). Алгоритмы ориентированные на шум типа SPN обладают высокой эффективностью, так как основаны на априорной информации о цвете поврежденных пикселей. Однако эти алгоритмы характеризуются не стопроцентной эффективностью в обнаружении поврежденных пикселей. Для импульсного шума случайного значения алгоритмы в большинстве случаев строятся на основе схемы SD-ROM [20-23]. Данная схема впервые была предложена в статье [20]. Основная ее идея состоит в анализе пикселей, расположенных в скользящем окне размером 3x3, и окружающих исследуемый пиксель. Решение о поврежденности пикселя принимается на основе пороговой схемы.

Целью данной статьи является разработка и апробация алгоритма выявления и устранения импульсного шума на изображениях на основе построения ассоциативных правил. Метод построения ассоциативных правил [24], являющийся одним из методов Data Mining, получил широкое распространение в планировании и организации продаж. Основная цель данного метода состоит в выявлении взаимосвязей между объектами на основе большого объема статистических данных. Рассматривая изображения как список небольших последовательностей пикселей, можно выявить закономерности, характерные для данного изображения. Выявление поврежденных пикселей можно проводить, выявляя явные отклонения от ожидаемых закономерностей. Заполнение пропущенных пикселей можно строить на основе наиболее часто встречающихся последовательностей. Предлагаемый в данной статье новый подход на основе метода ассоциативных правил обладает рядом преимуществ по сравнению с используемыми ранее алгоритмами. Во-первых, решение принимается на основе анализа всего изображения в целом, а не только локального окружения одного пикселя, что позволяет более полно учитывать основные закономерности расположения графических объектов. Во-вторых, рассматривается цепочки последовательных пикселей и выбираются из них наиболее вероятные для данного изображения, что позволяет анализировать протяженные объекты.

1. Постановка задачи

Будем представлять исходное неповрежденное изображение A в виде матрицы цветов A_{ij}. Если изображение цветное, то будем анализировать отдельно три матрицы, соответствующих модели RGB. Пусть диапазон значений элементов матрицы задан интервалом целых чисел $[0,m]$. Поврежденное импульсным шумом изображение будем задавать матрицей цветов B_{ij}. Пусть поврежденные импульсным шумом пиксели определяются матрицей повреждений R. Каждый элемент матрицы повреждений R_{ij} с вероятностью p отличен от нуля и с вероятностью $1-p$ равен нулю. Значение ненулевых элементов матрицы повреждений задается случайным образом в интервале $[0,m]$. Значение элементов матрицы поврежденного изображения B_{ij} задается соотношением

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

Поставим задачу поиска матрицы R_{ij} по известной матрице B_{ij}. Также будем искать наилучшее приближение C_{ij} матрицы цветов A_{ij} по известной матрице B_{ij}.

Для сравнения близости матриц изображений использовалась метрика Минковского [25, 26], согласно которой расстояние между изображениями находится по формуле

$$d(A,C) = \max_{n,m} \sum_{k=1}^{N} \frac{1}{N} \mid A_{nm}^{(k)} - C_{nm}^{(k)} \mid,$$

где A_{mn} и C_{mn} - значения цветов пикселов изображения A и C, N - количество пикселов.

2. Алгоритм поиска поврежденных пикселей

Как уже было сказано в постановке задачи, необходимо построить алгоритм поиска матрицы R по известной матрице B. Будем исходить из предположения о локальных закономерностях значений цветов изображения, а именно, из предположения, что существуют закономерности, связывающие каждый элемент матрицы с его ближайшими соседями. Причем эти закономерности являются свойствами всего изображения в целом. Поэтому, собрав информацию о взаимном расположении пикселей с различными значениями цвета, мы можем попытаться найти закономерности. Зная основные закономерности различного размещения цветов, можно выявить те пиксели, которые выпадают из найденных зависимостей. С большой вероятностью данные пиксели окажутся поврежденные импульсным шумом. Найдя все такие пиксели, мы получим матрицу R.

Одним из способов поиска закономерностей в статистических данных является метод построения ассоциативных правил. Для применения данного метода необходимо вы-

делить транзакции, содержащие записи, между которыми ищутся зависимости. Одной из главных проблем применения метода поиска ассоциативных правил является правильное формирование транзакций. В нашем случае в качестве транзакций будем использовать значения цветов k пикселей расположенных последовательно на одной прямой по вертикали или по горизонтали. Оптимальное значение k, позволяющее осуществлять поиск поврежденных пикселей с наибольшей точностью, должно зависеть от структуры изображения, а именно, от количества мелких деталей. Для каждого пикселя одна транзакция будет представлять собой набор из k-l-1 пикселей, расположенных левее него, и l пикселей, расположенных правее, либо из k-l-1 пикселей, расположенных выше, и l пикселей, расположенных ниже (l пробегает значения от 0 до k-1). Таким образом, для каждого пикселя строится 2k транзакций. Каждую транзакцию рассматриваем как набор из k пикселей, обозначаемый в дальнейшем F_i, и используемый для построения ассоциативных правил. Далее, для каждого набора цветов пикселей F_i определяем величину поддержки (support)

$$Supp_min(F_i) = \frac{D(F_i)}{N_1},$$

где $D(F_i)$ - количество транзакций, содержащих набор F_i; N_1 - общее количество транзакций. Из полученных наборов строим ассоциативные правила для каждого из встречающегося света пиксела c_j в наборе F_i:

$$\{c_1, c_2, ..., c_{j-1}, c_{j+1}, ..., c_k\} \Rightarrow c_j .$$

Для задачи поиска поврежденных пикселей интерес представляют те случаи, которые явно выпадают из общей закономерности. Очевидно, что в случае пикселя, явно выпадающего по цвету из своего окружения, будет получено ассоциативное правило с низкой поддержкой. Следует помнить, что с каждым пикселем связано 2k ассоциативных правил. Поврежденным будем считать тот пиксель, для которого уровень поддержки всех связанных с ним ассоциативных правил ниже некоторого порогового значения *Supp_min*.

При реализации предложенного алгоритма следует учитывать, что существует большое количество оттенков каждого цвета. В силу этого, может сложиться ситуация, при которой все ассоциативные правила имеют низкую поддержку. Поэтому, необходимо рассматривать цвет каждого пикселя с точностью до некоторой глубины. В нашей работе при построении ассоциативных правил у значения цвета каждого пикселя отбрасывались младшие b бит, что позволило близкие цвета считать одинаковыми.

Реализация предложенного алгоритма состоит из двух этапов. Во-первых, необходимо построить ассоциативные правила. Первый этап имеет линейную трудоемкость, од-

нако требует большого объема памяти. Уменьшить требования к памяти можно существенно, если сразу задать граничное значение *Supp_min* и хранить только ассоциативные правила, имеющие меньшую поддержку. На втором этапе необходимо осуществить повторный проход по изображению и выявление испорченных пикселей. Этот проход также имеет линейную сложность. Таким образом, при достаточном объеме памяти алгоритм будет работать быстро.

3. Компьютерный эксперимент по выявлению поврежденных пикселей

Для исследования эффективности алгоритма поиска поврежденных пикселей был проведен компьютерный эксперимент на трех типах изображений: однородной заливке, искуственном изображении геометрических фигур и изображении "Lena". Все изображения имели размер 256×256 пикселей, глубина цвета составляла 24 бита. С помощью генератора псевдослучайных последовательностей с равномерным распределением выбирались координаты поврежденного пикселя и его новый цвет. Исследования проводились при различной величине относительного количества поврежденных пикселей p от 10% до 70 %. Для того, чтобы не учитывать оттенки, отбрасывались младшие b=4 бита в каждом цветовом канале. Работа алгоритма оценивалась по двум параметрам: *eff* - эффективность, как количество правильно найденных поврежденных пикселей к общему количеству сгенерированных поврежденных пикселей; *err* - процент ложных срабатываний, как отношение ошибочно определенных пикселей к общему количеству пикселей, найденных алгоритмом.

На рисунке 1 представлен равномерно закрашенный фон с импульсным шумом интенсивности p=10%, а также сгенерированная карта поврежденных пикселей и найденные карты поврежденных пикселей при *Supp_min*=0.01, k=3 и k=5. Как показали численные оценки, эффективность обнаружения составила *eff*=99,7%, а ложные срабатывания *err*=0,4%. Были обнаружены не все поврежденные пиксели в связи с тем, что цвет их задавался случайно и некоторые из них имели цвет, близкий к фоновому. Наличие ложных срабатываний обусловлено попаданием поврежденных пикселей в близлежащие точки изображения.

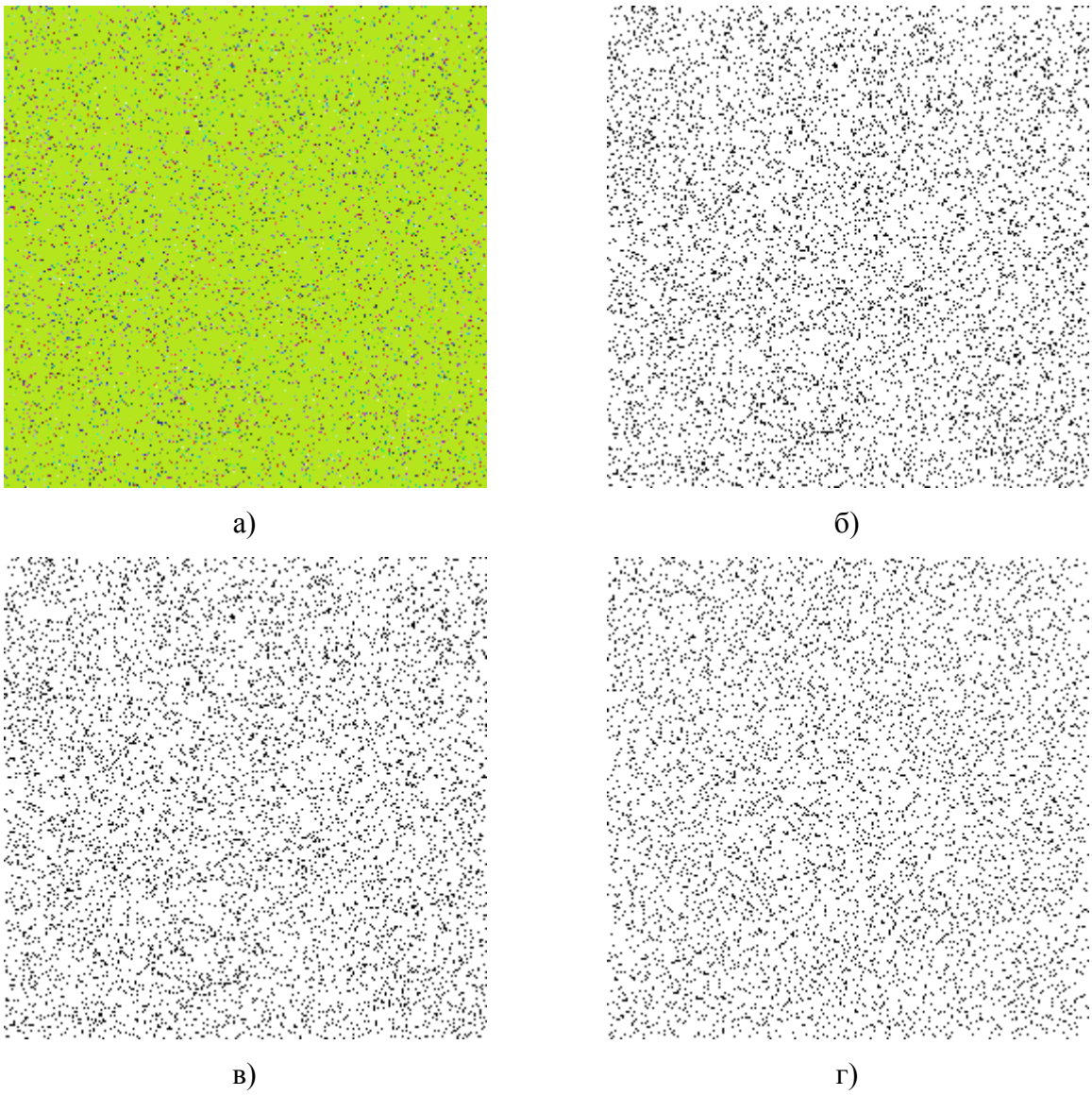
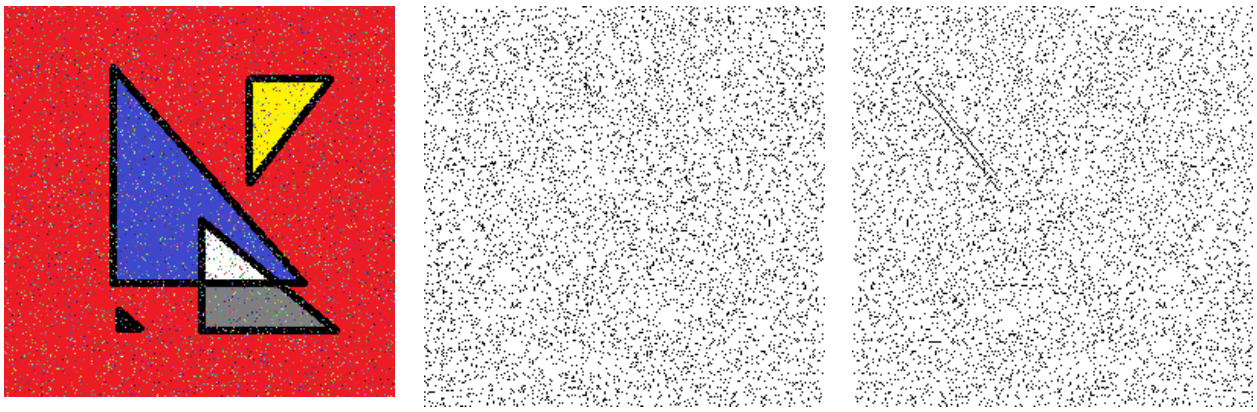


Рис. 1. Результат работы алгоритма по поиску поврежденных пикселей на изображении со сплошной заливкой с интенсивностью повреждений *p*=10%: а) зашумленное изображение, б) оригинальная карта поврежденных пикселей, в) карта поврежденных пикселей при *k*=3 и *Supp_min*=0,01, г) карта поврежденных пикселей при *k*=5 и *Supp_min*=0,01.

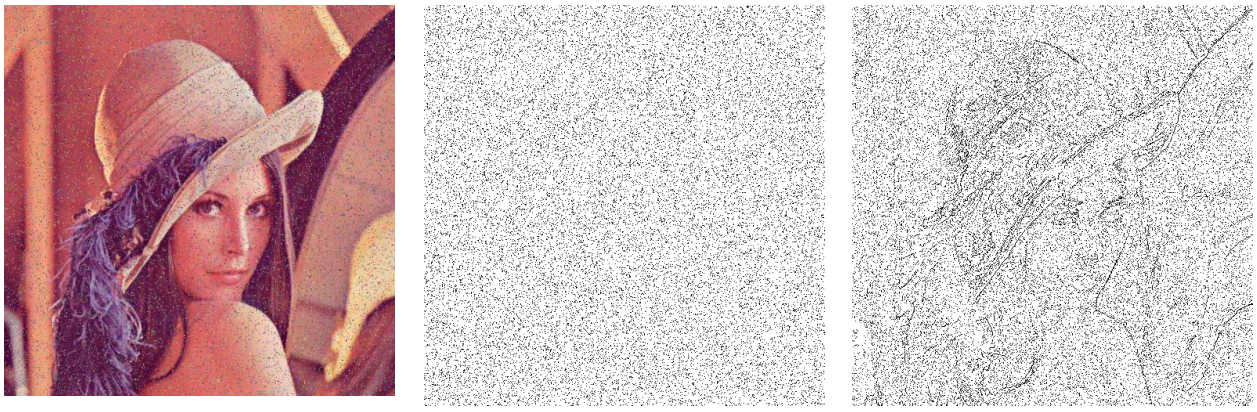
Более интересные результаты были получены для искусственного изображения геометрических фигур с четкими границами. Результаты представлены на рисунке 2 для случая *p*=10%, *k*=3, *Supp_min*=0,01. Эффективность определения поврежденных пикселей составила eff=96,.1%, ложные срабатывания составили err=3%. Как видим, при наличии резких переходов алгоритм определяет их как поврежденные пиксели. Данный эффект является ожидаемым, так как резкие границы содержат небольшое количество пикселей, и, как следствие, соответствующие им правила, имеют низкую поддержку.



а) б) в)

Рис. 2. Результат работы алгоритма по поиску поврежденных пикселей на изображении с геометрическими фигурами с интенсивностью повреждений *p*=10%: а) зашумленное изображение, б) оригинальная карта поврежденных пикселей, в) карта поврежденных пикселей, построенная алгоритмом при *k*=3 и *Supp_min*=0,01.

Результаты работы на реальном изображении "Lena" представлены на рисунке 3.



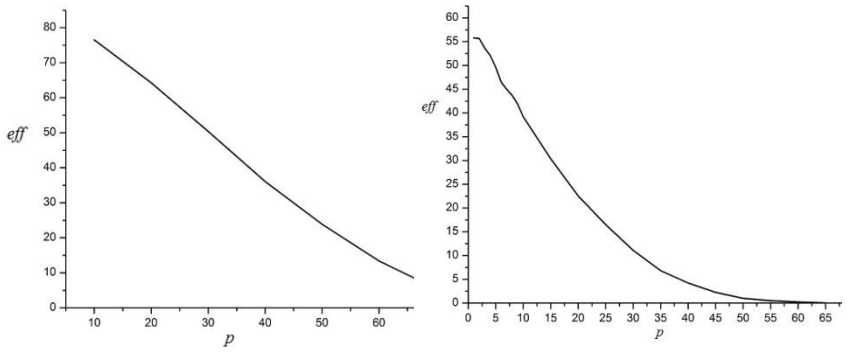
а) б) в)

Рис. 3. Результат работы алгоритма по поиску поврежденных пикселей на изображении "Lena" с интенсивностью повреждений *p*=10%: а) зашумленное изображение, б) исходная карта поврежденных пикселей, в) карта поврежденных пикселей при *k*=3 и *Supp_min*=0,01.

Эффективность работы алгоритма равна *eff*=76,9%, ложные срабатывания составили *err*=34,7%. Как видим, на реальном изображении достаточно высок процент ложных срабатываний. Это связано с тем, что в выбранном изображении "Lena" много мелких деталей, которые воспринимаются алгоритмом как поврежденные пиксели. Также на восстановленном изображении хорошо просматриваются контуры изображения.

Как было видно из рисунка 3, при увеличении количества цветов в палитре изменяется как эффективность обнаружения поврежденных пикселей, так и процент ложных

срабатываний. На рисунке 4 представлены зависимости эффективности работы алгоритма *eff* от уровня загрязнения изображения *p* при *k*=3 и *k*=5. Как видно из рисунка, эффективность работы алгоритма убывает при увеличении процента испорченных пикселей. Данный эффект вполне объясним. Уменьшается количество ассоциативных правил, присущих самому изображению, и растет количество ассоциативных правил, присущих шуму. Вследствие этого шум перестает восприниматься как отклонение от закономерностей и не детектируется.

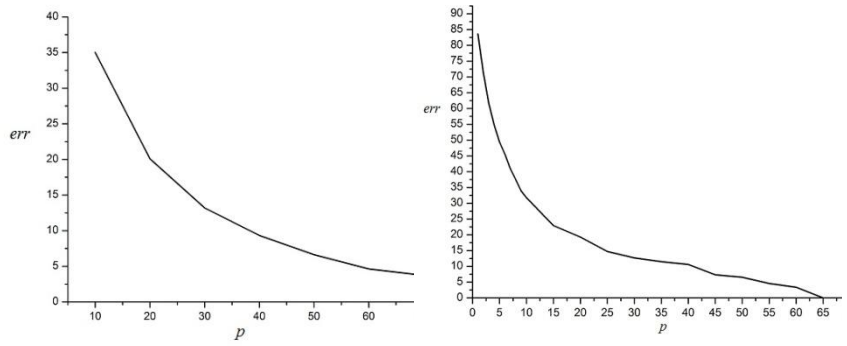


а) б)

Рис. 4. Зависимость эффективности работы алгоритма *eff* от уровня загрязнения изображения *p*: а) *k*=3, *Supp_min*=0,01, б) *k*=5, *Supp_min*=0,01.

Также из рисунка 4 хорошо видно, что эффективность снижается при увеличении длины набора *k*. Данный эффект связан с тем, что обилие мелких деталей не позволяет устанавливать взаимосвязи между пикселями, достаточно удаленными друг от друга.

На рисунке 5 представлены зависимости процента ложных срабатываний *err* от уровнях загрязнения *p*.

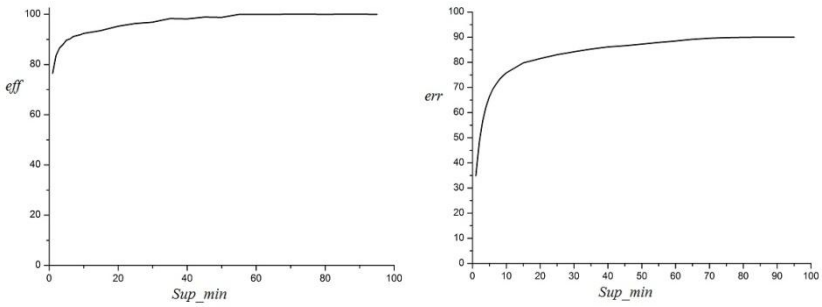


а) б)

Рис. 5. Зависимость процента ложных срабатываний *err* от уровня загрязнения *p*: а) *k*=3, *Supp_min*=0,01, б) *k*=5, *Supp_min*=0,.01.

Как видно из рисунка 5, процент ложных срабатываний уменьшается с увеличением величины импульсного шума. Данная закономерность является следствием того, что уменьшается общее количество пикселей, детектируемых как шум, так как поврежденных пикселей становится слишком много и алгоритм воспринимает их как часть исходного изображения.

Важным является выбор минимального уровня поддержки *Supp_min*. На рисунке 6 представлены зависимости эффективности работы алгоритма *eff* и процента ложных срабатываний *err* от минимального уровня поддержки *Supp_min* при *p*=10% и *k*=3.



а) б)

Рис. 6. Зависимость эффективности работы алгоритма *eff* (а) и процента ложных срабатываний *err* (б) от минимального уровня поддержки *Supp_min* при *p*=10% и *k*=3.

Как видно из графиков на рисунке 6, с ростом величины минимальной поддержки увеличивается как эффективность обнаружения поврежденных пикселей, так и количество неверно идентифицированных пикселей. Эта закономерность объясняется тем, что при высоком уровне минимальной поддержки мелкие детали, встречающиеся редко, воспринимаются алгоритмом как поврежденные пиксели.

При реализации прикладной системы идентификации поврежденных пикселей необходимо выбирать оптимальные параметры системы *Supp_min* и *k*. Однако в данном случае мы имеем дело с задачей многокритериальной 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 [27] при построении целевой функции:

$$F(k, Sup_min) = a_1 eff(k, Sup_min) + \frac{a_2}{err(k, Sup_min)} \to \min.$$

Положительные коэффициенты a_1 и a_2 выбираются из соображений значимости того или иного критерия и удовлетворяют условию $a_1+a_2=1$. Функции *eff* и *err* должны быть нормированы.

4. Алгоритм восстановления поврежденных пикселей

Рассмотрим вторую часть алгоритма восстановления изображения, поврежденного импульсным шумом. Будем использовать сформированные ассоциативные правила восстановления найденных испорченных пикселей. Для этого осуществим проход по всем пикселям, найденным предыдущим алгоритмом. Для каждого пикселя рассмотрим наборы троек подряд идущих пикселей, включающих в себя поврежденный пиксель и расположенных с ним на одной линии по вертикали или по горизонтали. Для каждой тройки (c_i,c_j,c_0), где c_0 - поврежденный пиксель, c_i и c_j - известные пиксели, рассмотрим все возможные ассоциативные правила $(c_i,c_j)=>c_k$. Пусть уровень поддержки этого правила d_k. Тогда в качестве цвета восстановленного пикселя может быть выбран либо цвет с наибольшей поддержкой

$$c_0 = \max_k d_k ,$$

Либо может быть выбрана взвешенная сумма всех ассоциативных правил

$$c_0 = \frac{\sum_k d_k c_k}{\sum_k d_k} .$$

Первый подход позволяет более четко восстанавливать границы контуров, второй подход лучше восстанавливает области с градиентной заливкой.

Как уже говорилось, на работу алгоритма существенное влияние оказывает выбор метода сравнения цветов. При большой глубине цвета на изображение будет присутствовать множество оттенков, что приведет к ситуации отсутствия повторений в наборах цветов. Используемое обрезание младших битов приводит к уменьшению набора цветов, из которого происходит выбор цвета восстановленного пикселя. Однако данное огрубление изображения практически не заметно визуально, особенно при использовании взвешенной суммы.

5. Компьютерный эксперимент по восстановлению поврежденных изображений

Компьютерный эксперимент проводился с растровыми изображениями размером 256x256 пикселей с глубиной цвета 8 бит. В качестве исходного объекта было выбрано

изображение "Lena". Основная схема эксперимента состояла в том, что на исходном изображении генерировался случайный импульсный шум. После чего собиралась статистика по неиспорченным пикселям и формировались ассоциативные правила с уровнем поддержки не ниже заданного значения *Supp_min*. Затем определялось положение поврежденных пикселей алгоритмом описанным выше. После этого осуществлялся проход по испорченным пикселам и их восстановление. Визуально эксперимент показал приемлемые результаты, представленные на рисунке 7.

а) б) в)

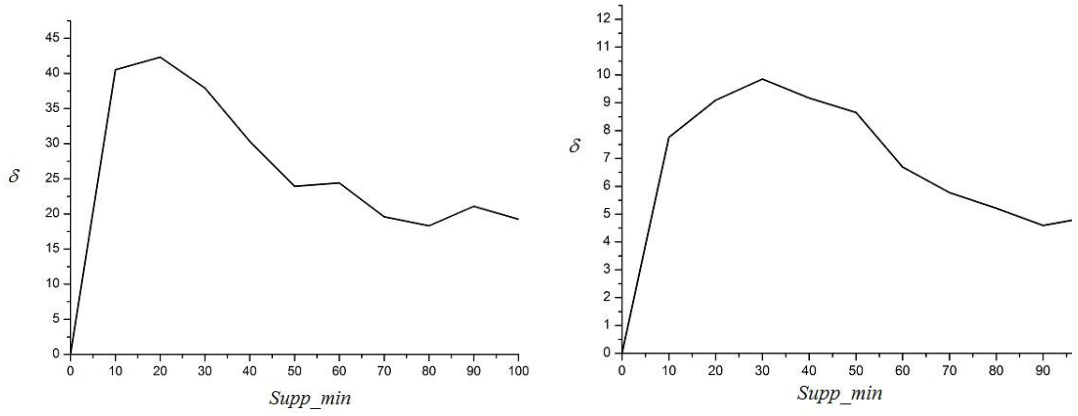
Рис. 7. Результат работы алгоритма по восстановлению пропущенных пикселей с *Supp_min*=0.2: а) исходное изображение, б) изображение, зашумленное на 10%, в) восстановленное изображение.

Предложенный метод восстановления изображений со статическими пропусками содержит ряд параметров, влияющих на качество восстановления. К таким параметрам можно отнести уровень поддержки *Supp_min* и размер транзакции *k*. В дальнейшем, при постановке компьютерного эксперимента, статистика собиралась для уровня шума от 0% до 70% с шагом 10, длина транзакции *k* бралась равной 3, 4 или 5. Эксперименты проводились для уровней поддержки от 0 до 100.

Прежде всего, была исследована зависимость результатов работы метода от порогового значения поддержки *Supp_min*. Относительное улучшение изображения вычислялось на основе расстояния *d(orig_fig,r_fig)* от восстановленного изображения *r_fig* до исходного *orig_fig* и расстояния *d(orig_fig, p_fig)* от испорченного изображения *p_fig* до исходного изображения *orig_fig*:

$$\delta = \frac{d(orig_fig, p_fig) - d(orig_fig, r_fig)}{d(orig_fig, p_fig)} \cdot 100\%.$$

Результаты эксперимента для уровня шума 10% и 50% представлены на рисунке 8. Для других значений уровня шума наблюдались аналогичные зависимости.



а) б)

Рис. 8. Зависимость относительного улучшения изображения от порогового уровня поддержки *Supp_min* и длины набора *k*=3 для уровня шума: а) 10% б) 50%.

Как хорошо видно из рисунка 8, эффективность работы алгоритма существенно зависит от порогового уровня поддержки. При всех значениях шума на графике присутствует максимум между значениями Supp_min от 20 до 30. При уровне шума 10% максимальное относительное улучшение изображения равно 42,3%, при уровне шума 50% - максимальное улучшение 9.8%. Поэтому в дальнейшем будем использовать значение *Supp_min*=20. Как видим при поиске и при восстановлении изображения необходимо выбирать различные уровни минимальной поддержки.

Исследуем зависимость эффективности восстановления от длины набора *k*. В таблице 1 приведены значения относительного улучшения при различных значениях *k* и нескольких значениях уровня шума.

Таблица 1. Значения относительного улучшения изображения *eff* при различных значениях длины набора *k* и различных уровнях шума *p*, *Supp_min*=20.

δ	***k*=3**	***k*=4**	***k*=5**
p=10%	42,30	28,87	15,17
p=30%	14,74	7,70	5,20
p=50%	9,85	4,56	2,06
p=70%	5,80	2,20	0,94

Как видно из таблицы 1, при увеличении длины набора *k* эффективность восстановления падает. Это связано с тем, что для изображений, содержащих достаточное количество деталей, накапливается большой объем различных видов наборов с низкой поддержкой. Причем, чем больше длина набора, тем больше различных видов наборов и тем ниже их поддержка.

Таким образом оптимальным является длина набора k=3 и минимальной поддержкой *Supp_min*=0,20. Рассмотрим зависимость относительного улучшения изображения от уровня зашумления при k=3 и *Supp_min*=0,20, представленную на рисунке 9. Как видно из рисунка, эффективность восстановления существенно снижается при увеличении зашумленности изображения, что является общим свойством всех существующих методов восстановления зашумленных изображений. Объясняется эта зависимость достаточно просто - чем меньше неиспорченных пикселей, тем меньше информации можно собрать для выработки ассоциативных правил.

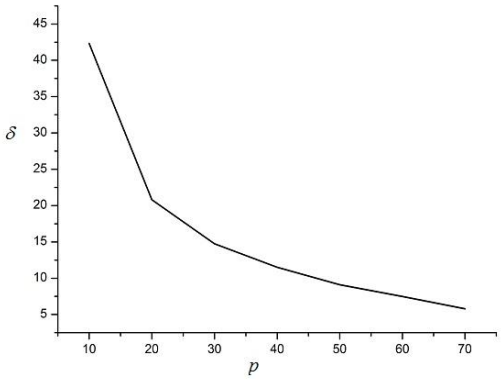
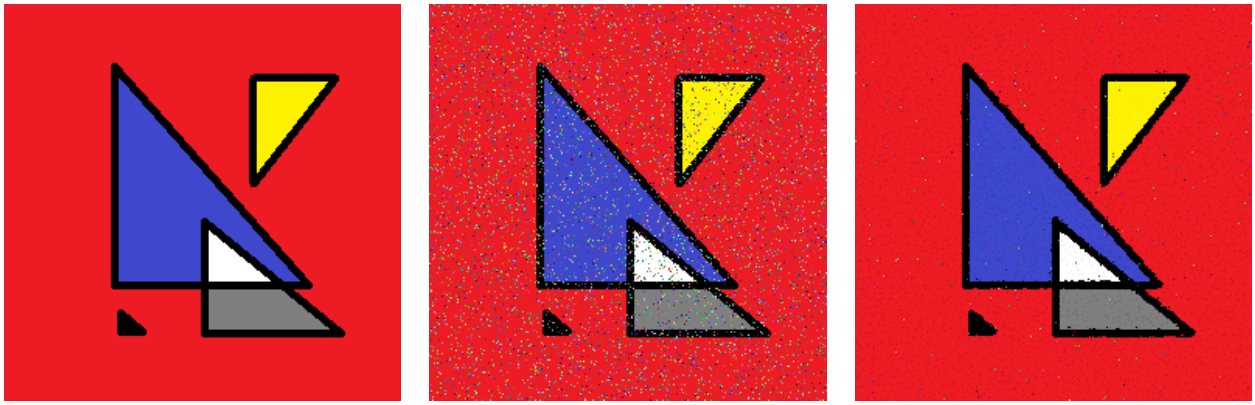


Рис. 9. Зависимость относительного улучшения изображения от уровня зашумления при k=3 и *Supp_min*=0,20.

Рассмотрим дополнительно вопрос о том, какие части изображения восстанавливаются лучше, а какие хуже. Для этого применим разработанный алгоритм к искусственному изображению с небольшим количеством деталей и четкими контурами. Результаты представлены на рисунке 10.



а) б) в)

Рис. 10. Результат работы алгоритма по восстановлению пропущенных пикселей для искусственного изображения: а) исходное изображение, б) изображение, зашумленное на 10%, в) восстановленное изображение.

Как видно из рисунка 10, алгоритм хорошо справляется с равномерно закрашенными областями и дает погрешности на резких границах и на краях изображения. Ошибки на резких границах возникают вследствие того, что количество наборов, соответствующих границе существенно уступает количеству наборов из сплошной области. Краевые эффекты обусловлены тем, что количество наборов, включающих испорченный пиксел, меньше чем у аналогов, удаленных от границы.

Не смотря на то, что согласно формальной метрике при больших уровнях зашумления изображений относительное увеличение не велико, визуально изображение улучшается значительно. Это связано с тем, что алгоритм находит решения близкие к правильным, но отличающиеся оттенками.

Заключение

Таким образом, алгоритмы поиска и восстановления поврежденных пикселей на основе формирования ассоциативных правил, предложенный в данной статье, дает достаточно хорошие результаты для не сильно зашумленных изображений. Эффективность алгоритма существенно зависит от выбора параметров формирования ассоциативных правил. При этом ассоциативные правила, сформированные на коротких цепочках, дают лучшие результаты, так как позволяют учитывать мелкие детали изображений, обрабатываемые неверно при увеличении длины цепочки.

Проведем сравнение полученных нами результатов с результатами других работ. В подавляющем числе случаев ставиться задача сглаживания импульсного шума и улучшение восприятия изображения без явного обнаружения испорченных пикселей. Ограничимся теми статьями, в которых постановка задачи совпадает с данной работой. В работе [29] проведено тестирование четырех различных алгоритмов обнаружения импульсного шума типа «соль и перец», для которых эффективность обнаружения поврежденных пикселей составляет от 71% до 96% при импульсном шуме с интенсивностью p=30%. Следует отметить, что решаемая в данной статье задача выявления импульсного шума случайного значения значительно более трудная по сравнению с выявлением SPN, однако предложенный метод на основе ассоциативных правил показывает сопоставимые по эффективности результаты.

Для методов выявления поврежденных пикселей для импульсного шума случайного значения на основе схемы SD-ROM [20] необходимо определение значений четырех параметров, играющих роль пороговых значений, которые выбираются пользователем произвольно из некоторого интервала и зависят от самого изображения. При правильном вы-

боре параметров алгоритм способен давать результаты сопоставимые с алгоритмами для шума типа «соль и перец». При ошибочном выборе параметров эффективность резко падает. Алгоритм, предложенный в данной статье, позволяет получать стабильные значения эффективности поиска поврежденных пикселей для различных изображений, не уступающий аналогичным работам, и не требует ручной подгонки параметров. Как хорошо известно, повреждения изображений импульсным шумом не являются критическим вплоть до значений *p* порядка 75% [29]. Предложенный алгоритм при таком уровне шума в качестве поврежденный 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 особенностью 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. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2005. 1072 с.
2. Pitas I., Venetsanopoulos A. Nonlinear Digital Filters: Principles and Applications. Springer US, 1990. DOI: 10.1007/978-1-4757-6017-0
3. Хуанг Т.С., Эклунд Дж.О., Нуссбаумер Г.Дж., Зохар Ш. Быстрые алгоритмы в цифровой обработке изображений: преобразования и медианные фильтры / под ред. Т.С. Хуанга; пер. с англ. под ред. Л.П. Ярославского. М.: Радио и связь, 1984. 224 с.
4. Sonka M., Hlavac V., Boyle R. Image Processing, Analysis, and Machine Vision. 1st ed. University Press, Cambridge, 2008. 920 p.
5. Mitra S., Sicuranza G. Nonlinear Image Processing. Academic Press, 2001. 455 p.
6. Buades A., Coll B., Morel J.-M. A non-local algorithm for image denoising // IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005). Vol. 2. IEEE Publ., 2005. P. 60-65. DOI: 10.1109/CVPR.2005.38
7. Dabov K., Foi A., Katkovnik V., Egiazarian K. Image denoising by sparse 3-D transform-domain collaborative filtering // IEEE Transactions on Image Processing. 2007. Vol. 16, no. 8. P. 2080-2095. DOI: 10.1109/TIP.2007.901238
8. Chan R., Ho C., Nikolova M. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization // IEEE Transactions on Image Processing. 2005. Vol. 14, no. 10. P. 1479 -1485. DOI: 10.1109/TIP.2005.852196

9. Kam H.S., Tan W.H. Noise detection fuzzy (NDF) filter for removing salt and pepper noise // In: Zaman H.B., Robinson P., Petrou M., Olivier P., Schröder H., Shih T.K., eds. Visual Informatics: Bridging Research and Practice. Springer Berlin Heidelberg, 2009. P. 479-486. (Ser. Lecture Notes in Computer Science; vol. 5857). DOI: 10.1007/978-3-642-05036-7_45
10. Ahamed J.N., Rajamani V. Design of hybrid filter for denoising images using fuzzy network and edge detecting // American Journal of Scientific Research. 2009. Iss. 3. P. 5-14.
11. Литл Р.Дж.А., Рубин Д.Б. Статистический анализ данных с пропусками: пер. с англ. М.: Финансы и статистика, 1990. 290 с.
12. Демиденко Е.З. Линейная и нелинейная регрессия. М.: Финансы и статистика, 1981. 302 с.
13. Ларионов И.Б. Восстановление изображений при помощи многомерных линейных многообразий // Проблемы обработки и защиты информации. Книга 2. Анализ графической и текстовой информации: коллективная монография / Под общей ред. д. ф.-м. н. С.В. Белима. Омск: ООО «Полиграфический центр КАН», 2010. С. 43-57.
14. Россиев А.А. Моделирование данных при помощи кривых для восстановления пробелов в таблицах // Методы нейроинформатики: сб. ст. / под ред. А.Н. Горбаня. Красноярск: КГТУ, 1998. С. 6-22 с.
15. Двоенко С.Д. Неиерархический дивизионный алгоритм кластеризации // Автоматика и телемеханика. 1999. № 4. С. 117-124.
16. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. Новосибирск: Изд-во ин-та математики, 1999. 270 с.
17. Загоруйко Н.Г., Елкина В.Н., Тимеркаев В.С. Алгоритм заполнения пропусков в эмпирических таблицах (алгоритм Zet) // Вычислительные системы: сб. тр. Вып. 61. Эмпирическое предсказание и распознавание образов. Новосибирск, 1975. С. 3-27.
18. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. М.: Горячая линия - Телеком, 2002. 382 с.
19. Ларионов И.Б. Карты Кохонена как способ восстановления мультимедийной информации // Журнал радиоэлектроники. 2010. № 10. Режим доступа: http://jre.cplire.ru/jre/oct10/3/text.html (дата обращения 01.10.2014).
20. 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. 1996. Vol. 5, no. 6. P. 1012-1025. DOI: 10.1109/83.503916
21. Garnett R., Huegerich T., Chui C., Wenjie He. A Universal Noise Removal Algorithm with an Impulse Detector // Transactions on Image Processing. 2005. Vol. 14, no. 11. P. 1747-1754. DOI: 10.1109/TIP.2005.857261
22. Красовский Г.Я., Усс М.Л. Фильтрация изображений, искаженных импульсными помехами точечного и строчного типа, на основе систем итерированных функций // Радіоелектронні і комп'ютерні системи. 2003. № 2. С. 47-55.

23. Сорокин С.В., Щербаков М.А. Реализация SD-ROM фильтра на основе концепции нечеткой логики // Известия высших учебных заведений. Поволжский регион. Технические науки. 2007. № 3. С. 56-65.
24. Agrawal R., Imieliński T., Swami A. Mining association rules between sets of items in large databases // SIGMOD '93 Proceedings of the 1993 ACM SIGMOD international conference on Management of data. New York: ACM Publ., 1993. P. 207-216. DOI: 10.1145/170035.170072
25. DiGesu V., Staravoitov V.V. Distance-based Functions for Image Comparison // Pattern Recognition Letters. 1999. Vol. 20, no. 2. P. 207-213. DOI: 10.1016/S0167-8655(98)00115-9
26. Jong-hyun Ryu, Sujin Kim, Hong Wan. Pareto Front Approximation with Adaptive Weighted Sum Method in Multiobjective Simulation Optimization // Proc. of the 2009 Winter Simulation Conference (WSC), TX, Austin, 2009. IEEE Publ., 2009. P. 623-633. DOI: 10.1109/WSC.2009.5429562
27. Ehrgott M. Multicriteria Optimization. Springer Berlin Heidelberg, 2005. 323 p. DOI: 10.1007/3-540-27659-9
28. Нгок Хоанг Ф., Спицын В.Г. Анализ алгоритмов обнаружения импульсного шума на цифровых изображениях // Известия Томского политехнического университета. 2011. Т. 318, № 5. С. 70-73.
29. Hutchinson J. Culture, Communication, and an Information age Madonna // IEEE Professional Communication Society Newsletter. 2001. Vol. 45, no. 3. P.1, 5-7.

Science & Education of the Bauman MSTU
Electronic journal
ISSN 1994-0408

Science and Education of the Bauman MSTU, 2014, no. 12, pp. 716–737.

DOI: **10.7463/1214.0744983**

Received: 13.07.2014
Revised: 18.11.2014

© Bauman Moscow State Technical Unversity

Algorithm for Searching the Broken Pixels and Eliminating Impulse Noise in Images Using a Method of Association Rules

S.V. Belim[1,*], A.O. Mayorov-Zilbernagel[1] *sbelim@mail.ru

[1]Omsk F.M. Dostoevskiy State University, Omsk, Russia

Keywords: data recovery, noisy images, association rules, the impulsive noise

This paper proposes an algorithm, which allows us to detect pixels broken by impulse noise on the basis of image analysis via a design method of association rules. A broken image is represented as a matrix. It is assumed that the matrix can be decomposed into the sum of two matrices. The first matrix is an intact image. The second matrix is a manifestation of impulse noise. There are two problems. It is necessary to define the non-zero elements of the matrix of noise. It is also required to restore the original image matrix as precisely as possible. The proposed algorithm consists of two stages. At the first stage the association rules are developed. As the transaction for association rules, the sets of adjacent pixels are used. The sets are formed from the entire image. Each set is considered as a set of pixels without due regard to the sequence order. The association rules are generated from the sets. The support level for each set is determined. The first stage has a linear complexity, but it requires large memory. At the second stage, there is repeated pass across the image and broken pixels are detected.

The proposed approach also allows us to recover broken images on the basis of association rules and damage masks detected at the first stage. For each spoiled pixel concerning its new color a decision is made on the basis of the association rules corresponding to the neighbors vertically and horizontally. As a new color, one is chosen that corresponds to the association rules with the greatest support.

The computer experiment is conducted to determine the effectiveness of the proposed method. A dependence of the algorithm efficiency on the various parameters is shown. The search efficiency for broken pixels is over 98% for artificial images and at least 76.9% for the natural ones. For images with many small details, there is rather high percentage of false positives. The best results can be attained with a set length of three pixels.

References

1. Gonzales R.C., Woods R.E. *Digital Image Processing*. 2nd ed. Prentice-Hall, 2001. (Russ ed.: Gonzales R., Vuds R. *Tsifrovaia obrabotka izobrazhenii*. Moscow, Tekhnosfera, 2006. 1072 p.)
2. Pitas I., Venetsanopoulos A. *Nonlinear Digital Filters: Principles and Applications*. Springer US, 1990. DOI: 10.1007/978-1-4757-6017-0
3. Huang T.S., ed.; Eklundh J.-O., Huang T.C., Justusson B.I., Nussbaumer H.J., Tyan S.G., Zohar S. *Two-Dimensional Digital Signal Prcessing II. Transforms and Median Filters*. Springer Berlin Heidelberg, 1981. DOI: 10.1007/BFb0057592 (Russ. ed.: Huang T.S., ed.; Eklundh J.-O., Nussbaumer H.J., Zohar S. *Bystrye algoritmy v tsifrovoy obrabotke izobrazheniy: preobrazovaniya i mediannye fil'try*. Moscow, Radio i Svyaz' Publ., 1984. 224 p.).
4. Sonka M., Hlavac V., Boyle R. *Image Processing, Analysis, and Machine Vision*. 1st ed. University Press, Cambridge, 2008. 920 p.
5. Mitra S., Sicuranza G. *Nonlinear Image Processing*. Academic Press, 2001. 455 p.
6. Buades A., Coll B., Morel J.-M. A non-local algorithm for image denoising. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005). Vol. 2*. IEEE Publ., 2005, pp. 60-65. DOI: 10.1109/CVPR.2005.38
7. Dabov K., Foi A., Katkovnik V., Egiazarian K. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Transactions on Image Processing*. 2007, vol. 16, no. 8, pp. 2080-2095. DOI: 10.1109/TIP.2007.901238
8. Chan R., Ho C., Nikolova M. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Transactions on Image Processing*, 2005, vol. 14, no. 10, pp. 1479 -1485. DOI: 10.1109/TIP.2005.852196
9. Kam H.S., Tan W.H. Noise detection fuzzy (NDF) filter for removing salt and pepper noise. In: Zaman H.B., Robinson P., Petrou M., Olivier P., Schröder H., Shih T.K., eds. *Visual Informatics: Bridging Research and Practice*. Springer Berlin Heidelberg, 2009, pp. 479-486. (Ser. *Lecture Notes in Computer Science*; vol. 5857). DOI: 10.1007/978-3-642-05036-7_45
10. Ahamed J.N., Rajamani V. Design of hybrid filter for denoising images using fuzzy network and edge detecting. *American Journal of Scientific Research*, 2009, is. 3, pp. 5-14.
11. Little R.J.A., Rubin D.B. *Statistical Analysis with Missing Data*. New York, John Wiley and Sons, 1987. (Russ. ed.: Little R.J.A., Rubin D.B. *Statisticheskiy analiz dannykh s propuskami*. Moscow, Finansy i statistika Publ., 1990. 290 p.).
12. Demidenko E.Z. *Lineynaya i nelineynaya regressiya* [Linear and nonlinear regression]. Moscow, Finansy i statistika Publ., 1981. 302 p. (in Russian).
13. Larionov I.B. Image restoration using multivariate linear manifolds. In: Belim S.V., ed. *Problemy obrabotki i zashchity informatsii. Kniga 2. Analiz graficheskoy i tekstovoy informatsii* [Problems of processing and protection of information. Book 2. Analysis of graphical and textual information]. Omsk, Publishing Center KAN, 2010, pp. 43-57. (in Russian).

14. Rossiev A.A. Data simulation using curves to fill the gaps in the tables. In: Gorban' A.N., ed. *Metody neyroinformatiki: sb. st.* [Methods of neuroinformatics: coll. art.]. Krasnoyarsk, KSTU Publ., 1998, pp. 6-22. (in Russian).
15. Dvoenko S.D. A nonhierarchical divisive clustering algorithm. *Avtomatika i telemekhanika*, 1999, no. 4, pp. 117-124. (English translation: *Automation and Remote Control*, 1999, no. 4, part 2, pp. 586-591.).
16. Zagoruyko N.G. *Prikladnye metody analiza dannykh i znaniy* [Applied methods of analysis of data and knowledge]. Novosibirsk, Publishing house of Institute of Mathematics, 1999. 270 p. (in Russian).
17. Zagoruyko N.G., Elkina V.N., Timerkaev V.S. Algorithm for filling gaps in empirical tables (algorithm Zet). *Vychislitel'nye sistemy: sb. tr. Vyp. 61. Empiricheskoe predskazanie i raspoznavanie obrazov* [Computing systems: coll. papers. Vol. 61. Empirical prediction and pattern recognition]. Novosibirsk, 1975, pp. 3-27. (in Russian).
18. Kruglov V.V., Borisov V.V. *Iskusstvennye neyronnye seti. Teoriya i praktika* [Artificial neural networks. Theory and practice]. Moscow, Goryachaya liniya – Telekom Publ., 2002. 382 p. (in Russian).
19. Larionov I.B. Kohonen maps as a way to restore the multimedia information. *Zhurnal radioelektroniki*, 2010, no. 10. Available at: http://jre.cplire.ru/jre/oct10/3/text.html , accessed 01.10.2014. (in Russian).
20. 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*, 1996, vol. 5, no. 6, pp. 1012-1025. DOI: 10.1109/83.503916
21. Garnett R., Huegerich T., Chui C., Wenjie He. A Universal Noise Removal Algorithm with an Impulse Detector. *IEEE Transactions on Image Processing*, 2005, vol. 14, no. 11, pp. 1747-1754. DOI: 10.1109/TIP.2005.857261
22. Krasovskiy G.Ya., Uss M.L. Based on systems of iterating functions method for filtration of images distorted by impulse noise of the type point and line. *Radioelektronni i komp'yuterni sistemi*, 2003, no. 2, pp. 47-55. (in Russian).
23. Sorokin S.V., Shcherbakov M.A. Implementing a CD-ROM filter based on the fuzzy logics concept. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Tekhnicheskie nauki = University proceedings. Volga region. Technical sciences*, 2007, no. 3, pp. 56-65. (in Russian).
24. Agrawal R., Imieliński T., Swami A. Mining association rules between sets of items in large databases. *SIGMOD '93 Proceedings of the 1993 ACM SIGMOD international conference on Management of data*. New York, ACM Publ., 1993, pp. 207-216. DOI: 10.1145/170035.170072
25. DiGesu V., Staravoitov V.V. Distance-based Functions for Image Comparison. *Pattern Recognition Letters*, 1999, vol. 20, no. 2, pp. 207-213. DOI: 10.1016/S0167-8655(98)00115-9

26. Jong-hyun Ryu, Sujin Kim, Hong Wan. Pareto Front Approximation with Adaptive Weighted Sum Method in Multiobjective Simulation Optimization. *Proc. of the 2009 Winter Simulation Conference (WSC)*, TX, Austin, 2009. IEEE Publ., 2009, pp. 623-633. DOI: 10.1109/WSC.2009.5429562
27. Ehrgott M. *Multicriteria Optimization*. Springer Berlin Heidelberg, 2005. 323 p. DOI: 10.1007/3-540-27659-9
28. Ngok Khoang F., Spitsyn V.G. Analysis of algorithms of detection of impulse noise in digital images. *Izvestiya Tomskogo politekhnicheskogo universiteta = Bulletin of the Tomsk Polytechnic University*, 2011, vol. 318, no. 5, pp. 70-73. (in Russian).
29. Hutchinson J. Culture, Communication, and an Information age Madonna. *IEEE Professional Communication Society Newsletter*, 2001, vol. 45, no. 3, pp.1, 5-7.