

Доклад

Тимонин А.С.

Тема: Методы и алгоритмы очистки и изображения от шумов

Удаление шума из изображений на основе нелинейных алгоритмов с использованием ранговой статистики

Аннотация

В последние два десятилетия в цифровой обработке изображений активно развиваются нелинейные алгоритмы на основе ранговой статистики для восстановления изображений, поврежденных различными моделями шумов. Подобные алгоритмы позволяют избежать дополнительного искажения изображения при удалении шума, а также значительно улучшить результаты работы фильтров на изображениях с высокой степенью зашумленности. В работе проводится сравнительный анализ как алгоритмов, использующих ранговую статистику, так и классических алгоритмов (медианный фильтр, фильтр Винера). Рассматриваются как объективные (численные) так и визуальные критерии сравнения результатов работы алгоритмов.

Ключевые слова: нелинейная фильтрация, ранжирующий фильтр, медианный фильтр, удаление шума, ранговая статистика, импульсный шум, восстановление изображений.

1. ВВЕДЕНИЕ

На практике часто встречаются изображения, искаженные

шумом, появляющимся на этапах формирования и/или передачи. Причинами возникновения шума на изображении могут быть сбои в работе канала связи, шум видеодатчика, дефект

пленки и др. [1]. Фундаментальной проблемой в области обработки изображений является эффективное удаление шума при сохранении важных для последующего семантического описания/распознавания деталей изображения. Сложность решения данной задачи существенно зависит от рассматриваемой модели шума. Анализ научно-технической литературы показывает, что наиболее адекватными с точки зрения использования в практических задачах являются модели аддитивного Гауссова и импульсного шума [2].

Аддитивный Гауссов шум характеризуется добавлением к каждому пикселю изображения значений из соответствующего нормального распределения с нулевым средним значением. Такой шум обычно вводится на этапе формирования цифровых изображений. Классические линейные фильтры, такие как фильтр Винера, способны эффективно удалить подобный шум, но степень размытости мелких деталей на изображении при этом может превысить допустимые задачей значения. Для решения этой проблемы используются нелинейные методы, например алгоритмы на основе анизотропной диффузии Перрона и Малика [3-5]. Другим интересным подходом является использование билатеральных и трилатеральных фильт-

величины [8]. Такая модель шума связана, например, с ошибками при передаче изображений. Для удаления импульсного шума используется специальный класс нелинейных фильтров, построенных на основе ранговой статистики [9]. Общей идеей таких фильтров является детектирование позиции импульса и замена его оценочным значением, при сохранении остальных пикселей изображения неизменными [10].

Целью работы является сравнение эффективности работы разработанного ранжирующего фильтра (РФ), билатерального и трилатерального фильтров с известными аналогами: медианным, фильтром Винера и модифицированным переключающимся прогрессивным медианным фильтром (МППМФ) [11]. Все рассматриваемые фильтры были реализованы в единой программной среде PicLab [12]. В качестве сравнительных критериев работы алгоритмов рассматривались:

– пиковое отношение сигнал/шум (ПОСШ)

$$ПОСШ = 20 \cdot \log_{10} \frac{\sqrt{255}}{\sqrt{CKO}}, \quad (1)$$

$$CKO = \frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2, \quad (2)$$

ров, предложенных в работах Томази и Мандуччи [6], базирующихся на оригинальной идее Овертона и Веймута [7]. Суть таких методов заключается в использовании локальных оценок, адекватных определению контура на изображении и сглаживания таких участков в наименьшей степени.

где x_i и y_i – значения пикселей исходного и восстановленного изображений соответственно, а СКО – среднеквадратичная ошибка;

– универсальный индекс качества (УИК) [13, 14] $-1 < UIK < 1$;
– относительное время работы (за единицу принималось время работы классического медианного фильтра с маской 3×3).

В качестве тестовых использовались стандартные изображения, такие как «Lenna», «Baboon», «Barbara», «Boat», «Elaine», «Peppers» и другие [12]. Проводилась параметрическая оптимизация всех рассматриваемых фильтров по критерию минимума ПОСШ.

2. РАНЖИРУЮЩИЙ ФИЛЬТР

2.1 Определение и структура

Рассмотрим вещественную двумерную последовательность $\{x(\mathbf{n})\}$ и определим $w(\mathbf{n})$ – как восьмиэлементный вектор, который содержит элементы из окна 3×3 , центрированного вокруг $x(\mathbf{n})$ (исключая сам пиксель $x(\mathbf{n})$) так, что:

$$w(\mathbf{n}) = [x_1(\mathbf{n}), x_2(\mathbf{n}), \dots, x_8(\mathbf{n})]. \quad (3)$$

В этом случае x_i соответствует отображению слева-направо и сверху-вниз окна 3×3 в одномерный вектор, как показано на рис. 1а. Элементы данного вектора могут быть также упорядочены в ряд, который определяется вектором

$$r(\mathbf{n}) = [r_1(\mathbf{n}), r_2(\mathbf{n}), \dots, r_8(\mathbf{n})], \quad (4)$$

где $r_1(\mathbf{n}), r_2(\mathbf{n}), \dots, r_8(\mathbf{n})$ – элементы $w(\mathbf{n})$, расположенные в возрастающем порядке, таком, что $r_1(\mathbf{n}) \leq r_2(\mathbf{n}) \leq \dots \leq r_8(\mathbf{n})$. Определим ранжированное (упорядоченное) среднее как

$$m(\mathbf{n}) = \frac{r_4(\mathbf{n}) + r_5(\mathbf{n})}{2}. \quad (5)$$

Заметим, что ранжированное среднее близко к определению медианы для девятиэлементного окна. Важное отличие

состоит в том, что $w(\mathbf{n})$ не включает центральный пиксель исходного окна 3×3 .

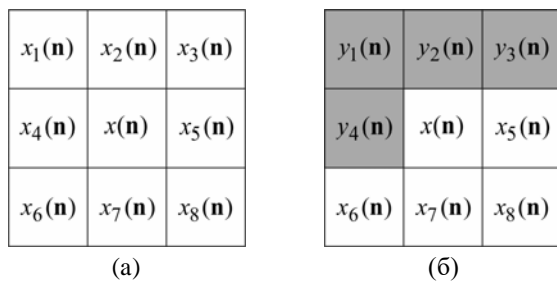


Рис. 1. Элементы скользящего окна размера 3×3 с центром в пикселе $x(\mathbf{n})$ для (а) рекурсивного случая, (б) нерекурсивного случая

Хотя алгоритмы в этом разделе определяются для окна 3×3 , в принципе, они могут быть распространены на окна произвольной формы и размера, а также применены для одномерных сигналов.

Определим классификатор C , который действует на текущее окно и распределяет его по категориям в один из множества классов M . Определим состояние переменной

$s(\mathbf{n}) = C(x(\mathbf{n}), w(\mathbf{n}))$ как результат этого классификатора и

будем использовать его для управления операцией фильтрации F , определённой в виде:

$$\begin{aligned} y(\mathbf{n}) &= F(x(\mathbf{n}), w(\mathbf{n}), s(\mathbf{n})) = \\ &= \alpha_{s(\mathbf{n})} x(\mathbf{n}) + \beta_{s(\mathbf{n})} m(\mathbf{n}), \end{aligned} \quad (6)$$

где $\{\alpha_i; i = 1, \dots, M\}$ и $\{\beta_i; i = 1, \dots, M\}$ – скалярные коэффициенты, соответствующие каждому из возможных M состояний. Функция переменной состояния нужна для вычисления вероятности наличия импульсного шума так, что коэффициенты $\{\alpha_i\}$ и $\{\beta_i\}$ могут быть выбраны с достаточной точностью для аппроксимации реального значения пикселя. Рассмотрим классификатор, который действует на основе разности между входным пикселем $x(\mathbf{n})$ и элементами в векторе $r(\mathbf{n})$. Эти ранжированные разности определяются как

$$d(\mathbf{n}) = \clubsuit r_k(\mathbf{n}) - x(\mathbf{n}), x(\mathbf{n}) \leq m(\mathbf{n}) \quad (7)$$

для ♥ 9 k

селя. Например, если разность $d_1(\mathbf{n})$ положительна, то текущий пиксель $x(\mathbf{n})$ имеет либо наибольшее, либо наименьшее значение в рассматриваемом окне. Если $d_1(\mathbf{n})$ не только положительна, но и достаточно велика по величине, то текущий пиксель очень похож на импульс. В совокупности, разности

$d_1(\mathbf{n}), \dots, d_4(\mathbf{n})$ дают намного больше информации о присут-

ствии повреждённых пикселей, даже для случая, когда в окне есть различные импульсы. Блок-схема структуры фильтра представлена на рис. 2.

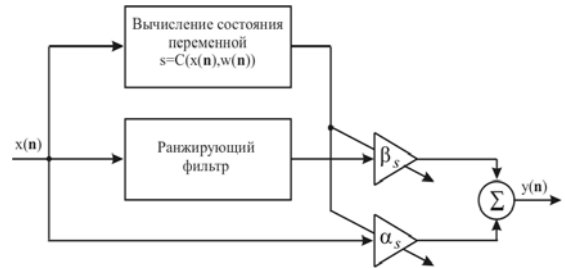


Рис. 2. Блок-схема ранжирующего фильтра

РФ обеспечивает лучшее восстановление, чем обычный медианный фильтр, особенно на изображениях сильно повреждённых импульсным шумом. Заметим, что процесс фильтрации (б) не является частным случаем фильтрации на основе ранговой статистики, поскольку окно $w(\mathbf{n})$ не включает центральный пиксель $x(\mathbf{n})$.

Во многих случаях улучшение работы может быть получено, если реализовать алгоритм в рекурсивной форме. Для

этого скользящее окно $w(\mathbf{n})$ должно содержать шумовые и

отфильтрованные пиксели, как показано на рис. 1б. Для рекурсивного случая скользящее окно переопределяется следующим образом:

$$w(\mathbf{n}) = [y_1(\mathbf{n}), \dots, y_4(\mathbf{n}), x_5(\mathbf{n}), \dots, x_8(\mathbf{n})], \quad (8)$$

где $y_i(\mathbf{n})$ соответствует выходному значению фильтра для

каждого шумового пикселя $x_i(\mathbf{n})$.

Ограничим каждую пару весовых коэффициентов α_i и β_i так, чтобы их суммы были нормализованы к единице. Эти ограничения выражаются как

$$\beta_i = 1 - \alpha_i, \quad i = 1, \dots, M, \quad (9)$$

что упрощает процесс, так как только один коэффициент нуждается в оптимизации в каждом состоянии. Это ограничение может быть также рассмотрено как достаточное условие для несмещённой оценки постоянного сигнала в присутствии аддитивного шума.

Рассмотрим, как ранжированные разности могут быть совместно использованы для эффективного удаления импульсного шума.

$k = 1, \dots, 4$

k

2.2 Метод двух состояний

♦ $x(\mathbf{n})$ $d_i(\mathbf{n}) \leq d$ В случае метода двух состояний цель классификатора заключается в определении, является ли текущий пиксель поврежденным или нет.

(за мет им, что, по опр еде лен ию,

для всех \mathbf{n} и $i < j$). Рассмотренный вектор разностей несет в себе информацию о вероятности повреждения текущего пик-

состояний цель классификатора за- ключается в определении, является ли текущий пиксель по- врежденным или нет. Решение принимается, используя ряд пороговых значений $T_1 < T_2 < T_3 < T_4$, исходя из которого

алгоритм считает $x(n)$ шумом и присваивает $s(n) = 1$, если

$$\text{хоть одно из неравенств верно: } d(n) > T_k, \quad k = 1, \dots, 4. \quad (10)$$

В противном случае алгоритм считает текущий пиксель $x(n)$ неповрежденным и устанавливает $s(n) = 2$.

Для реализации РФ необходимо задать пороги фильтрации T_1, T_2, T_3, T_4 .

На множестве исследованных изображений лучшие результаты были получены при пороговых значениях выбранных из следующего множества:

$$T_1 \in \{4, 8, 12\}, T_2 \in \{15, 20, 25\}, T_3 = 40, T_4 = 50.$$

Алгоритм работает хорошо даже для субоптимально-выбранных пороговых значений. На самом деле исходные значения

$$T_1 = 8, T_2 = 20, T_3 = 40, T_4 = 50$$

способны обеспечить отличные результаты на большинстве изображений поврежденных импульсными шумами как с фиксированными $[0, 255]$, так и со случайными (равномерное распределение) значениями пикселей. Основной результат моделирования заключается в том, что нет необходимости искать пороговые значения вне интервалов:

$$T_1 \leq 15, 15 \leq T_2 \leq 25, 30 \leq T_3 \leq 50, 40 \leq T_4 \leq 60.$$

Полученные оптимальные значения параметров РФ используются далее при проведении сравнительного анализа алгоритмов восстановления цифровых изображений.

3. РЕЗУЛЬТАТЫ МОДЕЛИРОВАНИЯ

Рассмотренный выше РФ был использован для удаления различных моделей шума из изображений. Результаты сравнивались как с широко известными алгоритмами фильтрации: фильтр Винера, медианный фильтр, так и с относительно новыми билатеральным и трилатеральным фильтрами. Численные оценки работы алгоритмов приведены в таблице 1. Кроме того, проводилась визуальная оценка восстановленных изображений (рис. 3). Результаты показывают, что

– на гауссовом шуме лучший результат показал билатераль-

ный фильтр. Трилатеральный фильтр и линейный фильтр Винера показали одинаковый результат, однако трилатераль-

ный значительно уступает в скорости работы, РФ практически не изменил изображения, оставив аддитивный гауссов шум без изменений;

– на импульсном шуме с фиксированными значениями импульсов с вероятностью появления импульсов $p = 5\%$ лучший результат показал РФ, примерно в два раза проиграв по времени МППМФ; детектор импульсов дает некоторое преимущество трилатеральному фильтру над медианным, но временные затраты остаются значительными; система детектирования билатерального фильтра распознает импульсный шум как границы контуров изображения и оставляет их неизменными;

– на импульсном шуме с равномерным распределением значений импульсов с вероятностью появления импульсов $p = 10\%$ картина остается примерно такой же как и при воздействии импульсным шумом с фиксированными значениями, за исключением того, что преимущество РФ становится более весомым.

4. ВЫВОДЫ

Анализ таблицы 1 показывает, что разработанный алгоритм РФ превосходит алгоритмы медианной и адаптивной фильтрации по критерию ПОСШ восстановленного изображения на 4-6 дБ.

Также установлено, что для модели импульсного шума с фиксированными значениями импульсов предложенный алгоритм РФ демонстрирует отличные результаты при малом уровне шума. Этот результат справедлив также и для импульсного шума со случайными значениями импульсов (равномерное распределение). РФ плохо справляется с гауссовым шумом и в случае такого повреждения целесообразнее применять билатеральный фильтр.

Трилатеральный фильтр хоть и проигрывает РФ и билатеральному фильтру, однако является более универсальным, так как неплохо справляется со всеми рассмотренными моделями шумов.

| Шум Тип фильтра | Гауссов шум ($m = 0, \sigma^2 = 100$) | | | Импульсный шум с фиксированными значениями импульсов | | | Импульсный шум со случайными значениями импульсов | | |
|-------------------------|---|-------------|----------------------------|--|-------------|----------------------------|---|-------------|----------------------------|
| | ПОСШ, дБ | УИК | Относительное время работы | ПОСШ, дБ | УИК | Относительное время работы | ПОСШ, дБ | УИК | Относительное время работы |
| Зашумленное изображение | 28,12 | 0,41 | – | 18,52 | 0,40 | – | 19,24 | 0,29 | – |
| Медианный | 32,17 | 0,49 | 1,00 | 34,73 | 0,66 | 1,00 | 33,87 | 0,65 | 1,00 |
| Винера | 33,61 | 0,54 | 1,79 | 17,66 | 0,28 | 1,52 | 19,32 | 0,27 | 1,63 |
| МППМФ | – | – | – | 41,70 | 0,93 | 1,13 | 24,44 | 0,58 | 1,12 |
| РФ | 28,81 | 0,42 | 2,45 | 42,53 | 0,98 | 2,62 | 37,23 | 0,92 | 2,75 |
| Билатеральный | 33,64 | 0,57 | 5,14 | 18,56 | 0,31 | 5,38 | 19,56 | 0,26 | 5,62 |
| Трилатеральный | 33,16 | 0,54 | 29,57 | 35,19 | 0,67 | 35,00 | 34,44 | 0,66 | 34,74 |

Табл. 1. Результаты фильтрации импульсного шума со случайными значениями импульсов

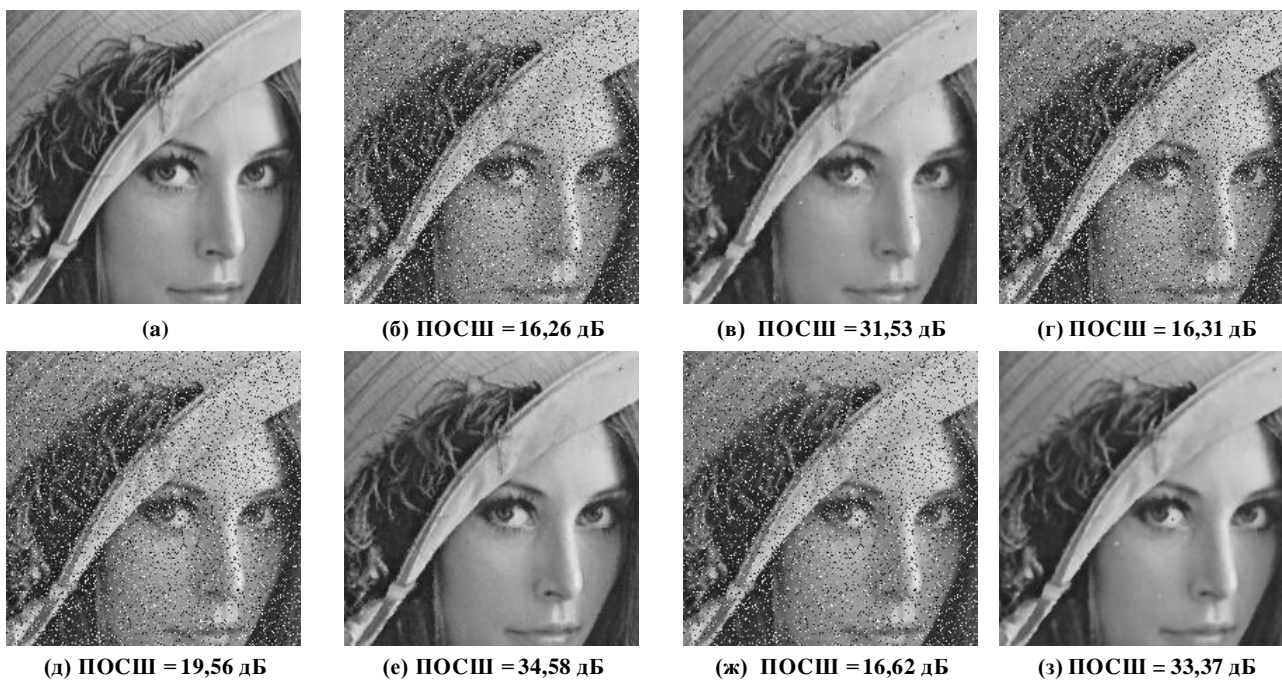


Рис. 3. Удаление импульсного шума со случайными значениями импульсов (увеличенный фрагмент изображения «Lenna»): (а) исходное тестовое изображение, (б) зашумленное изображение ($p = 20\%$), (в) результат обработки медианным фильтром, (г) результат обработки фильтром Винера, (д) результат обработки МППМФ фильтром, (е) результат обработки ранжирующим фильтром, (ж) результат обработки билатеральным фильтром, (з) результат обработки трилатеральным фильтром