

УДК 621.391

## ВЕСОВОЙ МЕТОД РЕШЕНИЯ ПРОБЛЕМЫ ГРАНИЧНЫХ ПИКСЕЛЕЙ В АЛГОРИТМЕ СВЕРТОЧНОЙ ФИЛЬТРАЦИИ ЦИФРОВОГО ШУМА

Д.В. ЗАЕРКО, В.А. ЛИПНИЦКИЙ

*Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь**Поступила в редакцию 30 марта 2020*

**Аннотация.** Описан весовой метод решения проблемы граничных пикселей, для алгоритмов использующих матрицы скручивания. Алгоритм рассмотрен в контексте применения матриц скручивания для подавления цифрового шума на растровых изображениях [1]. Алгоритм имеет ряд преимуществ перед широко используемыми. Он позволяет: отказаться от хранения временных изображений, использовать наиболее близкую цветовую гамму для граничных пикселей.

**Ключевые слова:** подавление цифрового шума, матрицы скручивания, операция свертки, граничные пиксели матрицы скручивания.

### Введение

Применение сверточной фильтрации для устранения цифрового шума на растровых изображениях широко распространено. Операция свертки [2, 3], как и применение матриц скручивания, является основой для множества методов таких как: линейное усреднение точек по соседям, гауссовское размытие и т.д.

Однако, при всех преимуществах использования алгоритма матриц скручивания существуют так же и «слабые места» алгоритма, прежде всего, его работа с граничными пикселями изображения [4]. Для корректной работы, алгоритму требуются значения дополнительных пикселей вне изображения. Существует множество методов для решения этой проблемы, но оптимального, с точки зрения использования ресурсов для обработки изображения и осуществляющих корректное цветовое наполнение граничных пикселей, пока нет. Это указывает на необходимость соблюдения консенсуса между экономией вычислительных ресурсов и улучшением обрабатываемого изображения. Алгоритм свертки, дополненный модификацией для корректной обработки граничных пикселей, не должен расточительно использовать ресурсы, но определять наиболее близкую цветовую гамму граничных пикселей относительно исходного изображения. Решению выявленной проблемы и посвящена данная работа.

### Весовая модификация сверточного алгоритма фильтрации цифрового шума

Рассмотрим альтернативный, весовой метод работы с граничными пикселями для алгоритма свертки. Модификация не предполагает создания промежуточного изображения, а лишь использования квадратной подматрицы пикселей  $\mathbf{A}^+ = \{a_{i,j}^+\}_{n \times n}$  с размерностью  $n$  и состоящей из  $n^2$  пикселей оригинальной матрицы  $\mathbf{A} = \{a_{i,j}\}_{h \times w}$ . Множества номеров строк и столбцов матрицы  $\mathbf{A}$  обозначим через  $I = \overline{0, h+1}, J = \overline{0, w+1}$ .

Очень важными является диапазоны изменения индексов строк и столбцов  $i, j$  (далее так же  $k, p$ ) элементов пиксельной подматрицы. Диапазоны изменений будут существенно различаться, в зависимости от расположения стороны (левая, нижняя, правая, верхняя) граничных пикселей относительно оригинальной пиксельной матрицы изображения  $\mathbf{A}$ . Расположение и соответствующие ему диапазоны индексов будут определяться, исходя из заданных номеров

строк и столбцов  $I = \overline{0, h+1}, J = \overline{0, w+1}$  граничного элемента матрицы, требующего вычисления по правилу:

$$(i, j) = \begin{cases} i = \overline{1+n(I-1), In}; j = \overline{1, n}; & \text{если, } J = 0; 0 < I < h+1; \\ i = \overline{(h+1)-n, h}; j = \overline{1+n(J-1), Jn}; & \text{если, } 0 < J < w+1; I = h+1; \\ i = \overline{1+n(I-1), In}; j = \overline{(w+1)-n, w}; & \text{если, } J = w+1; 0 < I < h+1; \\ i = \overline{1, n}; j = \overline{1+n(J-1), Jn}; & \text{если, } 0 < J < w+1; I = 0. \end{cases}$$

Далее указание изменения диапазонов строк и столбцов в описании формул будет опускаться, предполагая выбор его заранее в зависимости от  $I = \overline{0, h+1}, J = \overline{0, w+1}$ .

Цвет проблемного пикселя определяется, исходя из наиболее часто встречающегося цвета (по максимальному весу) среди других  $n^2$  пикселей  $a_{i,j}^+$  подматрицы  $\mathbf{A}^+$ , которые располагаются в непосредственной близости от граничного пикселя.

Под непосредственно близкими, имеется в виду  $n^2$  пикселей являющихся смежными к граничному пикселю на глубину до  $n$  пикселей.

Величина  $n$  определяется произвольно, исходя только из следующих двух условий.

1. Ограничения вычислительных возможностей системы.
2. Пиксельный размер изображения, ограничиваясь меньшей границей, т.е.

$$n = \begin{cases} 2, 3, 4, \dots, h-1; & h \leq w; \\ 2, 3, 4, \dots, w-1; & w \leq h. \end{cases}$$

Модификация предполагает использование специальных весовых коэффициентов пикселей  $\alpha_{i,j}$  для каждого пикселя, отвечающих за оценку «встречаемости» этого цвета среди других цветов  $n^2$  пикселей.

Итак, опишем основные этапы алгоритма, учитывая известные строку и столбец  $I = \overline{0, h+1}, J = \overline{0, w+1}$  граничного пикселя, для которого необходимо провести вычисление.

**1. Вычисление угловых точек.** Особым случаем будет расчёт четырех «угловых» граничных пикселей относительно оригинального изображения, которые могут быть вычислены сразу же, и из-за их небольшого числа не будут играть существенную роль. Они будут представлять собой копию ближайшего пикселя из подматрицы  $\mathbf{A}^+$  :

$$a_{0,0} = a_{1,1}^+; a_{h+1,0} = a_{h,1}^+; a_{0,w+1} = a_{1,w}^+; a_{h+1,w+1} = a_{h,w}^+.$$

**2. Подготовка весовых коэффициентов пикселей.** Все весовые коэффициенты  $\alpha_{i,j}$  для элементов  $a_{i,j}^+$  принимают равные значения  $\alpha_{i,j} = \frac{1}{n^2}$ .

**3. Вычисление результирующих весовых коэффициентов.** Для вычисления результирующего весового коэффициента  $res(\alpha_{i,j})$  для пикселя  $a_{i,j}^+$  необходимо суммировать весовые коэффициенты  $\alpha_{k,p}$ ; пикселей  $a_{k,p}^+$  при равенстве  $a_{i,j}^+ = a_{k,p}^+$  их пиксельного значения с пикселем  $a_{i,j}^+$ . Иначе оставить результирующий вес  $res(\alpha_{i,j})$  неизменным.

$$res(\alpha_{i,j}) = \sum_k \sum_p \{ \alpha_{k,p} \mid a_{i,j}^+ = a_{k,p}^+ \}.$$

**4. Определение значения цвета граничного пикселя.** Значения граничных пикселей определяются по оценке результирующих весовых коэффициентов, полученных на шаге 2.

Среди результирующих весовых коэффициентов  $res(\alpha_{i,j})$  для пикселей  $a_{i,j}^+$  вычисляется коэффициент с максимальным значением. Если его значение не равно заданному значению на

первом шаге  $\max\{res(\alpha_{i,j})\} \neq \frac{1}{n^2}$ , то очевидно, что коэффициент характеризует пиксель  $a_{i,j}^+$  с наиболее часто повторяемым цветом среди других  $n^2$  пикселей и граничный пиксель примет его значение. Если максимальное значение не изменилось:  $\max\{res(\alpha_{i,j})\} = \frac{1}{n^2}$ , то все  $n^2$  пиксели имеют различные цвета или сформировались группы пикселей, равные по числу пикселей, но различные по цвету. В этом случае граничным пикселем будет первый ближайший смежный пиксель, учитывая расположение граничного пикселя относительно стороны оригинального изображения, т.е:  $a_{\lfloor \frac{n}{2} \rfloor, 1}^+$  для левых граничных пикселей на изображении,  $a_{\lfloor \frac{n}{2} \rfloor, w}^+$  для правых,  $a_{1, \lfloor \frac{n}{2} \rfloor}^+$  для верхних,  $a_{h, \lfloor \frac{n}{2} \rfloor}^+$  для нижних. Представим вышесказанное в виде формул, подставляя вместо  $I = \overline{0, h+1}, J = \overline{0, w+1}$  номер строки и столбца для пикселя, для которого необходимо провести вычисление.

При  $J = 0; 0 < I < h+1$  для расчета левых граничных пикселей:

$$a_{I,0} = \begin{cases} a_{i,j}^+, & \text{если } \max\{res(\alpha_{i,j})\} \neq \frac{1}{n^2}; i = \overline{1+n(I-1), nI}; j = \overline{1, n}; \\ a_{\lfloor \frac{n}{2} \rfloor, 1}^+, & \text{если } \max\{res(\alpha_{i,j})\} = \frac{1}{n^2}; i = \overline{1+n(I-1), nI}; j = \overline{1, n}. \end{cases}$$

При  $0 < J < w+1; I = h+1$  для расчета нижних граничных пикселей:

$$a_{h+1,J} = \begin{cases} a_{i,j}^+, & \text{если } \max\{res(\alpha_{i,j})\} \neq \frac{1}{n^2}; i = \overline{(h+1)-n, h}; j = \overline{1+n(J-1), Jn}; \\ a_{\lfloor \frac{n}{2} \rfloor, 1}^+, & \text{если } \max\{res(\alpha_{i,j})\} = \frac{1}{n^2}; i = \overline{(h+1)-n, h}; j = \overline{1+n(J-1), Jn}. \end{cases}$$

При  $J = w+1; 0 < I < h+1$  для расчета правых граничных пикселей:

$$a_{I,w+1} = \begin{cases} a_{i,j}^+, & \text{если } \max\{res(\alpha_{i,j})\} \neq \frac{1}{n^2}; i = \overline{1+n(I-1), In}; j = \overline{(w+1)-n, w}; \\ a_{\lfloor \frac{n}{2} \rfloor, w}^+, & \text{если } \max\{res(\alpha_{i,j})\} = \frac{1}{n^2}; i = \overline{1+n(I-1), In}; j = \overline{(w+1)-n, w}. \end{cases}$$

При  $0 < J < w+1; I = 0$  для расчета верхних граничных пикселей:

$$a_{0,J} = \begin{cases} a_{i,j}^+, & \text{если } \max\{res(\alpha_{i,j})\} \neq \frac{1}{n^2}; i = \overline{1, n}; j = \overline{1+n(J-1), Jn}; \\ a_{\lfloor \frac{n}{2} \rfloor, 1}^+, & \text{если } \max\{res(\alpha_{i,j})\} = \frac{1}{n^2}; i = \overline{1, n}; j = \overline{1+n(J-1), Jn}. \end{cases}$$

Возможен так же альтернативный вариант весовому методу, предполагающий вычисление среднего значения цвета по  $n^2$  пикселям. Однако оба метода имеют проблемы при достаточно большом значении  $n$ .

В первом случае возникает проблема «слепого цветового дублирования», т.е. когда  $n$  достаточно велико и проблемный пиксель заполнится цветовым значением подавляющего из числа анализируемых, что может быть во всех случаях корректно. Во втором случае проблема возможна «цветового диссонанса», когда  $n$  достаточно велико и цвет проблемного пикселя будет «усреднен» и будет выделяться относительно граничных оригинального изображения.

### Заключение

В работе представлен весовой метод решения проблемы определения граничных пикселей. Он дополняет алгоритмы, использующие матрицы скручивания при подавлении цифрового шума на растровых изображениях. Метод не предполагает создания промежуточных изображений и организует цветовое наполнение пикселей в соответствии с наиболее используемым цветом на некоторой области на границе изображения. Метод не является оптимальным решением проблемы, однако дополняет уже существующий спектр методов, имея перед ними преимущество в простоте реализации, экономии ресурсов, а также в оптимальности цветового наполнения граничных пикселей.

### WEIGHTED METHOD SOLVING OF BOUNDARY PIXELS PROBLEM IN DIGITAL NOISE CONVOLUTION FILTERING ALGORITHM

D.V. ZAERKO, V.A. LIPNITSKY

**Abstract.** The weighted method solving of boundary pixels problem is described for algorithm which are used twist matrix. The algorithm is considered in the context of using twist matrices to suppress digital noise in bitmap images. The algorithm has several advantages over the widely used ones. Method allows to refuse to store temporary images, use the closest gamut colour for border pixels.

*Keywords:* digital noise reduction, twisting matrices, convolution operation, boundary pixels of a twist matrix

### Список литературы

1. Цветков В.Ю., Конопелько В.К., Липницкий В.А. Предсказание, распознавание и формирование образов многоакурсных изображений с подвижных объектов. Мн.: «Издательский центр БГУ». 2014.
2. Хиршман И.И., Уиддер Д.В. Преобразования типа свертки. М.: Издательство иностранной литературы, 1958.
3. Гашников М.В., Методы компьютерной обработки изображений / Под ред. В.А. Сойфера. 2-е изд., испр. М.: ФИЗМАТЛИТ. 2003.
4. Брейсуэлл Р.Н. Преобразование Хартли: Пер с англ. М.: Мир. 1990.