

## Реконструкционная прорисовка посредством восстановления градиентов изображения

В.Н. Карнаухов\*, В.И. Кобер\*, М.Г. Мозеров\*, Л.В. Зимина\*\*

\*Институт проблем передачи информации, Российская академия наук, Москва, 127051, Россия

\*\*Московский политехнический университет, Москва, 107023, Россия

Поступила в редколлегию 06.06.2022

**Аннотация**—Реконструкционная прорисовка используется как для улучшения визуального восприятия изображения, так и в классических задачах распознавания и робототехники с целью убрать нерелевантную информацию с изображения. Современные методы реконструкционной прорисовки используют нейронные сети. Однако у этих подходов существуют недостатки, не позволяющие использовать данные алгоритмы в практике компьютерного зрения. Авторы данной работы предложили (в их вышедшей недавно статье) метод прорисовки с использованием фильтра нелокального среднего (НЛС), который превосходит другие методы классической обработки изображения по критерию PSNR. Однако у этого алгоритма могут возникать артефакты на границах зон прорисовки, искажающие визуальное восприятие. В данной статье мы предлагаем прорисовывать не изображение, а градиенты восстанавливаемого изображения и только затем восстанавливать само изображение, таким образом сохраняя связанность на границе зон прорисовки. Этот метод преодолевает большую часть артефактов, возникающих при использовании предложенного ранее метода, а также превосходит его и по точности восстановления, выраженной по критерию PSNR.

**КЛЮЧЕВЫЕ СЛОВА:** Реконструкционная прорисовка, фильтр нелокального среднего, восстановление изображений, улучшения качества и контраста изображений.

DOI: 10.53921/18195822\_2022\_22\_2\_93

### 1. ВВЕДЕНИЕ

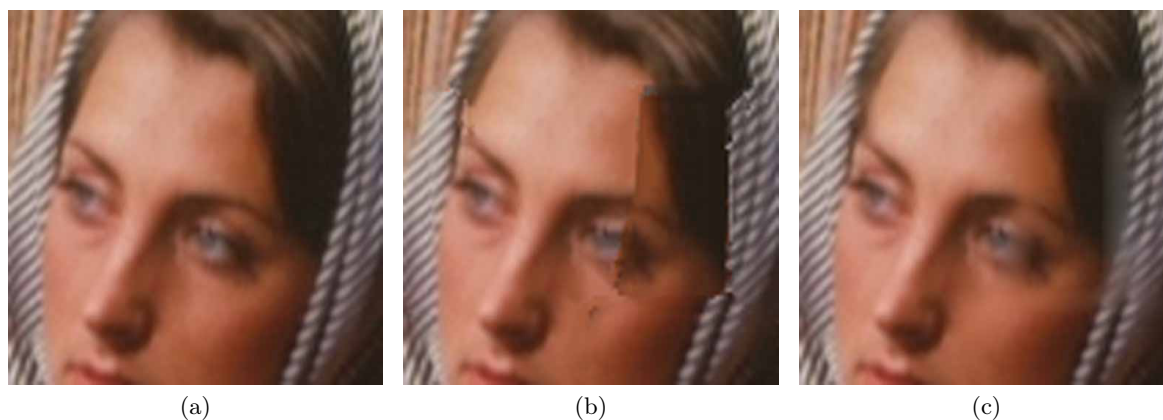
Одна из задач цифровой обработки изображений является улучшение качества деталей изображения для визуализации. Искажения, ухудшающие восприятие изображения, возникают по различным причинам, связанным с техническими характеристиками оптических систем, движением регистрирующей камеры или объекта и т.д. Похожая проблема реконструкционной прорисовки (impainting) в компьютерном зрении возникает в случае необходимости заполнить области изображения, искаженные артефактами, текстурой из неискаженных областей. Наиболее часто встречающимися артефактами является атрибутивный текст на изображениях и случайные объекты визуальной сцены, требующие удаления из поля зрения. Реконструкционная прорисовка используется как для улучшения визуального восприятия изображения, так и в классических задачах распознавания и робототехники с целью убрать нерелевантную информацию с изображения. В процессе удаления текста с изображений и видео часто возникает множество нежелательных визуальных артефактов. В решении этой проблемы важную роль играет классическая техника реконструкционной прорисовки [1–4].

В работе [5] представлено контекстно-зависимое отображение изображения на основе патчей с использованием фильтрации Габора и методов случайного поля Маркова (MRF). Однако скорость метода, как и других алгоритмов связанных с MRF, невысока и часто приводит к артефактам, связанным с глобальной минимизацией энергии.

Наиболее эффективные методы реконструкционной прорисовки (РП) используют нейронные сети [6, 7]. Однако у этих подходов существует как минимум три существенных недостатка: необходимость специального оборудования в виде GPU, гигантские объёмы информации кода (параметрические веса алгоритма достигают десятков ГБайт), зависимость от баз данных на которых производилось обучение конкретной нейронной сети.

Прогресс в восстановлении зашумленных изображений и улучшении качества деталей связан, в первую очередь, с появлением билатеральных фильтров [8]. В этом случае окрестность усреднения определяется не только пространственной близостью к восстанавливаемому пикселу, но и близостью в области значений. Такие фильтры широко используются в задачах обработки изображения и компьютерного зрения, таких как стерео [9, 10], стерео в движении [11, 12], оценка оптического потока [13], улучшение изображений и контрастирование [14–16], видео абстракция и устранение мозаичного эффекта demosaicing [17, 18]. Однако существенным недостатком фильтров с билатеральным ядром является квадратичная зависимость вычислительной сложности от числа пикселей на изображении. И хотя были предложены быстрые методы [19–22], они достаточно сложны в реализации, выход фильтра является неточной аппроксимацией истинного билатерального фильтра, и вычислительная сложность зависит от параметров фильтра.

Однако основным недостатком билатеральных фильтров в задачах РП является отсутствие знания о значении сигнала в точке восстановления. Альтернативное решение было представлено в статье [23], где было предложено использовать фильтр нелокального среднего (НЛС) [24], который является логическим продолжением подхода билатеральных фильтров. Для алгоритма НЛС близость окрестности значений расширяется до близости не только самого значения обрабатываемого пикселя (который считается неизвестным), но и всех других значений в небольшой пространственной окрестности этого пикселя. Такая окрестность называется заплатой (или patch по-английски). Качество восстановления НЛС алгоритма превзошло ранее разработанные алгоритмы и до сих пор входит в число лучших методов денойзинга [25–27]. В данной статье используют быструю реализацию фильтра НЛС, предложенную в [28]. Предложенный в статье [23] метод РП на основе фильтра НЛС превосходит наиболее простые и быстрые алгоритмы по критерию PSNR. Однако восстановление с помощью НЛС не лишено в некоторых случаях побочных артефактов, которые негативно сказываются на восприятии восстановленного изображения зрительной системой человека. Особенно если зона прорисовки обладает значительной шириной, как например на Рис. 1. Из этой иллюстрации видно, что



**Рис. 1.** Артефакты прорисовки методом [23]: (a) — зуммированное исходное изображение, (b) — результат прорисовки с НЛС, (c) — результат прорисовки с помощью метода [23].

внутренняя текстура зоны прорисовки отличается средним значением и таким образом создаются ложные границы восприятия, совпадающие с визуальными границами зоны прорисовки.

Идея данной статьи прорисовывать не изображение, а градиенты восстанавливаемого изображения, используя метод [23], а затем восстанавливать само изображение. Таким образом сохраняется связанность градиента на границе зон прорисовки, что не приводит к разрыву первого рода. В экспериментальной части показано, что предложенный метод преодолевает большую часть артефактов, возникающих при использовании метода [23], а кроме того, превосходит этот метод и по точности восстановления выраженный по критерию PSNR.

Статья организована следующим образом: в разделе 2 предложен алгоритм РП на основе фильтра НЛС, в разделе 3 приведены экспериментальные результаты и наконец, раздел заключение суммирует наши выводы.

## 2. МЕТОД НЕЛОКАЛЬНОГО СРЕДНЕГО АЛГОРИТМ ВОССТАНОВЛЕНИЯ ПО ГРАДИЕНТАМ

Фильтрация НЛС принадлежит к классу нелинейных фильтров усредняющих значения текущего зашумленного сигнала  $u_p$  по некоторой окрестности подобия следующим образом:

$$\tilde{u}_p = \frac{\sum_{q \in N_p^R} w_{p,q} u_q}{\sum_{q \in N_p^R} w_{p,q}}, \quad (1)$$

где  $u_p$  — исходный зашумленный сигнал,  $\tilde{u}_p$  — выход фильтра,  $p, q \in \mathcal{V}$  — множество пикселей (или вершин) и множество границ  $\varepsilon = (p, q) \in \mathcal{E}$  составляют граф изображения  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ ,  $N_p^R$  — нелокальная окрестность пикселя  $p$ . Знаменатель (1) нормализует выходной сигнал таким образом, чтобы области значения входного и выходного сигнала фильтра совпадали. Веса  $w_{p,q}$  отражают степень влияния значения  $u_q$  на выходное значение  $\tilde{u}_p$  и в случае линейной фильтрации такие веса зависят только от расстояния между пикселями  $p$  и  $q$ . Для НЛС фильтра веса  $w_{p,q}$  являются степенью близости значений входной функции  $u_p$  в некоторой окрестности пикселей  $p$  и  $q$ :

$$w_{p,q} = e^{-\frac{\|u(N_p^r) - u(N_q^r)\|_{2,a}^2}{h^2}}, \quad (2)$$

где условное расстояние по окрестности выражено следующей суммой:

$$\|u(N_p^r) - u(N_q^r)\|_{2,a}^2 = \sum_{\hat{p}, \hat{q} \in N_p^r, N_q^r} (u_{\hat{p}} - u_{\hat{q}})^2, \quad (3)$$

где окрестность  $N_p^r$  и есть заплата, по которой определяют близость пикселей  $p$  и  $q$ . Реализация этого фильтра для задачи РП принципиально не отличается от классической. Тем не менее необходимо описать эти отличия формально. Первое отличие касается определения двух окрестностей  $N_p^R$  в (1) и  $N_p^r$  в (3). Эти окрестности не должны включать точки подлежащие восстановлению. Второе отличие касается очередности восстановления точек на изображении. Прорисовка осуществляется не одновременно во все точках, о которых исходная информация отсутствует, а пошагово, как такая очередность шагов заполнения областей прорисовки показана на Рис. 2, где каждый шаг выделен своим цветом. Кроме того, после каждого шага заполнения, заполненные точки считаются информативными и их значения могут быть использованы в окрестностях  $N_p^R$  в (1) и  $N_p^r$  в (3). Зоны шагов прорисовки получаются в результате морфологической операции эрозии к первоначальной области, подлежащей прорисовке.

# ТЕКСТ

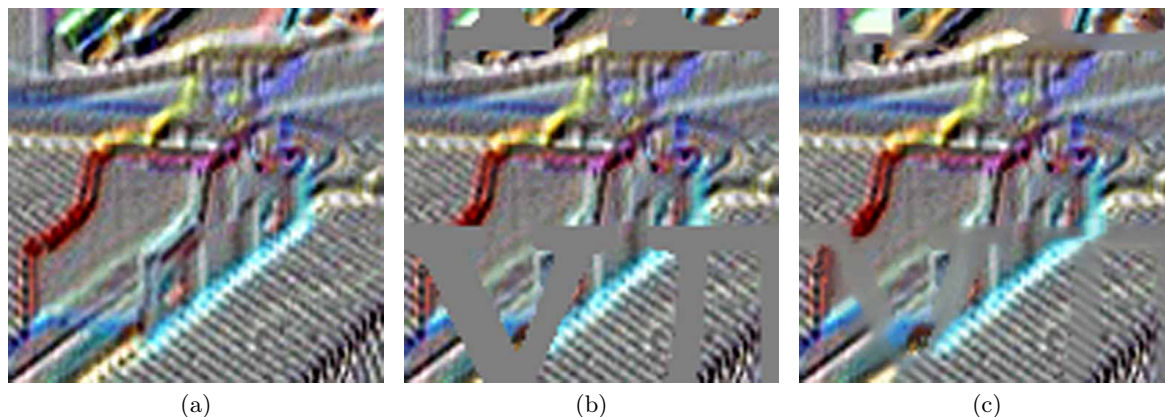
**Рис. 2.** Очередность шагов заполнения областей прорисовки. Каждый шаг выделен своим цветом.

Ключевая идея этой статьи прорисовывать не само изображение, а поле градиентов изображения, выраженных формулой:

$$\begin{aligned} g_x(x, y) &= u(x + 1, y) - u(x, y) \\ g_y(x, y) &= u(x, y + 1) - u(x, y), \end{aligned} \quad (4)$$

где  $(x, y)$  — координаты пиксела  $p$  на изображении,  $g_x$  и  $g_y$  — частные производные градиента изображения,  $u(x, y)$  — сигнал самого изображения.

Алгоритм прорисовки градиента изображения ничем принципиально не отличается от прорисовки самого изображения. Результат такой прорисовки показан на Рис. 3.



**Рис. 3.** Результат прорисовки градиентов исходного изображения: (a) — поле градиентов исходного изображения, (b) — маскированный градиент изображения, (c) — восстановленный градиент.

Теперь прорисовка собственно изображения заключается в итеративном выполнении условия локального градиента. А именно:

$$u(x, y) = \begin{cases} u(x + 1, y) - g_x(x, y) \\ u(x, y + 1) - g_y(x, y) \\ u(x, y - 1) + g_y(x, y - 1) \\ u(x - 1, y) + g_x(x - 1, y). \end{cases} \quad (5)$$

То есть значение сигнала  $u(x, y)$  в точке  $(x, y)$  можно вычислить четырьмя разными способами, используя четыре соседних значения, что следует из определения градиента в (4). Однако это условие выполняется только для собственного градиента изображения, и может не выполняться для прорисованного. Это известная проблема восстановления функции по искаженному градиенту (например в задаче *shape from shading* — поверхность по теням). Самый простой способ — это вычислить искомое значение, используя каждое из четырех окрестных, а затем

найти среднее по формуле:

$$u(x, y) = \frac{1}{4} \begin{pmatrix} u(x+1, y) - g_x(x, y) + \\ u(x, y+1) - g_y(x, y) + \\ u(x, y-1) + g_y(x, y-1) + \\ u(x-1, y) + g_x(x-1, y) \end{pmatrix}. \quad (6)$$

Если вычислить градиент восстановленного изображения по формуле (4) и обозначить его как  $\tilde{\mathbf{g}}$ , а затем вычислить ошибку  $E$  по формуле:

$$E = \sum_{p \in M} (\tilde{\mathbf{g}}_p - \mathbf{g}_p)^2, \quad (7)$$

где  $M$  — область прорисовки, то такая ошибка почти наверняка будет ненулевой. Поэтому восстановление по формуле (6) необходимо проводить итеративно, чтобы минимизировать ошибку. Более того, такая итерация должна быть последовательной, от пиксела к пикселу, чтобы ускорить сходимость. Критерием остановки будет некий порог разницы восстановленного участка изображения на шаге итерации  $N$  по сравнению с этим же участком на предыдущем шаге. В нашем случае разница выбрана в 70 dB по критерию PSNR.

### 3. ЭКСПЕРИМЕНТАЛЬНЫЕ РЕЗУЛЬТАТЫ

Экспериментальная секция нашей работы поделена на две части где:

- мы анализируем численные результаты и преимущества использования прорисовки градиентов над пропиской непосредственно изображения.
- часть результатов экспериментов представлена в иллюстрациях статьи чтобы читатель мог сравнить качество восстановления визуально.

#### 3.1. Эксперименты с численным анализом результатов

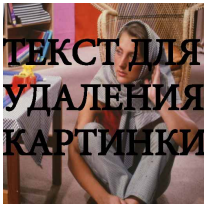
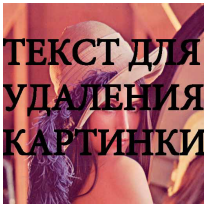
Здесь представлены результаты работы четырех алгоритмов, локального среднего (ЛС), алгоритм описанный в статье [4], метод статьи [23] и предложенный алгоритм с предварительной прорисовкой градиентов и применением фильтра НЛС. Результат, полученный с помощью предложенного метода сравнивается с восстановленным изображением методом локального среднего (ЛС) и методом описанным в статьях [4] и [23]. В Таблице 1 представлены результаты сравнения по критерию отношения сигнал/шум (PSNR) в dB по отношению к неискаженному изображению.

Анализируя результат в Таблице 1 мы можем сказать что предложенный в статье алгоритм достаточно сильно улучшает точность восстановления базового алгоритма [23]. Однако главное достоинство предложенного алгоритма — устранение артефактов, возникающих в базовом алгоритме [23]. В следующей подсекции мы подробнее опишем этот результат.

#### 3.2. Результаты экспериментов для визуального анализа

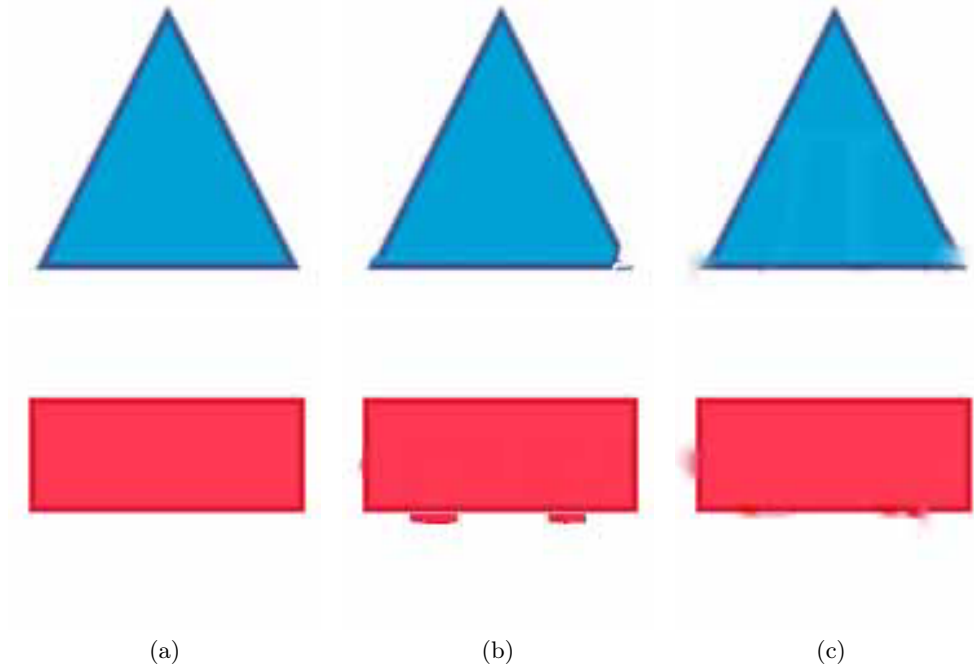
Устранение артефактов, возникающих в базовом алгоритме [23], показано на двух рисунках: Рис. 1 и Рис. 4. Иллюстрация показывает, что наш алгоритм успешно устраняет артефакты базового метода. Кроме того, на Рис. 4 видно, что новый алгоритм пытается достроить угол треугольника, что достигается именно восстановлением утраченных значения градиента изображения.

**Таблица 1.** Результаты сравнения по критерию отношения сигнал/шум (PSNR) в dB по отношению к неискаженному изображению. Здесь представлены результаты работы четырех алгоритмов, локального среднего (ЛС), методами описанным в статьях [4] и [23], и предложенный алгоритм с предварительной прорисовкой градиента (ПГ).

Изображения:	Barbara	Lena	Peppers	Geometry
Алгоритмы:				
ЛС	20.41	21.13	21.85	16.15
[4]	20.88	21.49	22.26	16.86
[23]	22.71	22.10	23.49	19.36
ПГ	23.17	22.73	23.78	19.86

Кроме того, в этой части экспериментальной секции, представлен результат прорисовки различными алгоритмами для визуального сравнения. Здесь представлены результаты работы четырёх алгоритмов, локального среднего (ЛС), алгоритм описанный в статье [4], методом [23], и предложенный алгоритм с применением прорисовки градиента(ПГ) фильтром НЛС. Результаты представлены на Рис. 5 для визуального сравнения.

Анализируя результат представленный на Рис. 5 мы можем сказать что предложенный в статье алгоритм РП лучше восстанавливает текстуру в областях с неизвестными значениями, если сравнивать с другими представленными методами.



**Рис. 4.** Артефакты прорисовки методом [23] и их устранение на примере изображений с геометрическими фигурами Geometry: (a) — зуммированное исходное изображение, (b) — результат прорисовки с НЛС, (c) — результат прорисовки с помощью метода [23].



**Рис. 5.** Результат прорисовки различными алгоритмами для визуального сравнения. Здесь представлены результаты работы четырёх алгоритмов: локального среднего (ЛС), алгоритм, описанный в статье [4], методом [23] и предложенный алгоритм с применением прорисовки градиента (ПГ) фильтром НЛС. НИИ — неискаженное исходное изображение; ИИ — искаженное изображение.

## ЗАКЛЮЧЕНИЕ

В этой работе мы предложили улучшенный вариант метода прорисовки на основе фильтра НЛС для задачи РП. Главная идея предложенного алгоритма, устранение артефактов возникающих в базовом алгоритме [23]. Результат компьютерных экспериментов показал, что предложенный метод реконструкционной прорисовки превосходит другие методы по критерию PSNR. Кроме того, достоинства применения предложенного фильтра в задаче РП также показано с помощью иллюстраций к статье, чтобы читатель мог сравнить качество различных вариантов обработки визуально.

## СПИСОК ЛИТЕРАТУРЫ

1. Hirani A. N., Totsuka T. Combining frequency and spatial domain information for fast interactive image noise removal // Proceedings of the 23rd annual conference on Computer graphics and interactive techniques. 1996. P. 269–276.
2. Efros A. A., Leung T. K. Texture synthesis by non-parametric sampling // Proceedings of the seventh IEEE international conference on computer vision / IEEE. Vol. 2. 1999. P. 1033–1038.
3. Heeger D. J., Bergen J. R. Pyramid-based texture analysis/synthesis // Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. 1995. P. 229–238.
4. Karras D., Mertzos G. Discretization schemes and numerical approximations of pde inpainting models and a comparative evaluation on novel real world mri reconstruction applications // 2004 IEEE International Workshop on Imaging Systems and Techniques (IST)(IEEE Cat. No. 04EX896) / IEEE. 2004. P. 153–158.
5. Ružić T., Pižurica A. Context-aware patch-based image inpainting using markov random field modeling // IEEE transactions on image processing. 2014. Vol. 24, no. 1. P. 444–456.
6. Demir U., Unal G. Patch-based image inpainting with generative adversarial networks // arXiv preprint arXiv:1803.07422. 2018.
7. Yeh R. A., Chen C., Yian Lim T. et al. Semantic image inpainting with deep generative models // Proceedings of the IEEE conference on computer vision and pattern recognition. 2017. P. 5485–5493.
8. Tomasi C., Manduchi R. Bilateral filtering for gray and color images // In CVPR. 1998. P. 839–846.
9. Mozerov M., van de Weijer J. Accurate stereo matching by two-step energy minimization // IEEE Transactions on Image Processing. 2015. Vol. 24, no. 3. P. 1153–1163.
10. Mozerov M. G., van de Weijer J. One-view occlusion detection for stereo matching with a fully connected crf model // IEEE Transactions on Image Processing. 2019. Vol. 28, no. 6. P. 2936–2947.
11. Ershov E., Karnaukhov V., Mozerov M. Probabilistic choice between symmetric disparities in motion stereo matching for a lateral navigation system // Optical Engineering. 2016. Vol. 55, no. 2. P. 023101–023101.
12. Mozerov M., van de Weijer J. Improved recursive geodesic distance computation for edge preserving filter // IEEE Transactions on Image Processing. 2017. Vol. 26, no. 8. P. 3696–3706.
13. Mozerov M. Constrained optical flow estimation as a matching problem // IEEE Transactions on Image Processing. 2013. Vol. 22, no. 5. P. 2044–2055.
14. Fattal R., Agrawala M., Rusinkiewicz S. Multiscale shape and detail enhancement from multi-light image collections // ACM Transactions on Graphics (TOG) / ACM. Vol. 26. 2007. P. 51.
15. Farbman Z., Fattal R., Lischinski D., Szeliski R. Edge-preserving decompositions for multi-scale tone and detail manipulation // ACM Transactions on Graphics (TOG) / ACM. Vol. 27. 2008. P. 67.
16. Karnaukhov V., Mozerov M. Restoration of multispectral images by the gradient reconstruction method and estimation of the blur parameters on the basis of the multipurpose matching model // Journal of Communications Technology and Electronics. 2016. Vol. 61, no. 12. P. 1426–1431.



17. Ramanath R., Snyder W. E. Adaptive demosaicking // *J. Electron. Imag.* 2003. Vol. 12, no. 4. P. 633–642.
18. Winnemöller H., Olsen S., Gooch B. Real-time video abstraction // *ACM Transactions on Graphics.* 2006. Vol. 25, no. 3. P. 1221–1226.
19. Paris S., Durand F. A fast approximation of the bilateral filter using a signal processing approach // *Proc. European Conf. on Computer Vision.* 2006. P. 568–580.
20. Adams A., Baek J., Davis M. Fast high-dimensional filtering using the permutohedral lattice // *Computer Graphics Forum.* 2010. Vol. 29, no. 2. P. 753–762.
21. Gastal E., Oliveira M. Domain transform for edge-aware image and video processing // *ACM Transactions on Graphics.* 2011. Vol. 30, no. 4. P. 69.
22. Mozerov M. G., van de Weijer J. Global color sparseness and a local statistics prior for fast bilateral filtering // *Image Processing, IEEE Transactions on.* 2015. Vol. 24, no. 12. P. 5842–5853.
23. Karnaukhov V., Kober V., Mozerov M., Zimina L. Impainting with a nonlocal means filter // *Journal of Communications Technology and Electronics.* 2022. Vol. 67, no. 6. P. 709–714.
24. Buades A., Coll B., Morel J.-M. A non-local algorithm for image denoising // *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on / IEEE.* Vol. 2. 2005. P. 60–65.
25. 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.
26. Zuo W., Zhang L., Song C., Zhang D. Texture enhanced image denoising via gradient histogram preservation // *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2013. P. 1203–1210.
27. Liu G., Zhong H., Jiao L. Comparing noisy patches for image denoising: A double noise similarity model // *IEEE Transactions on Image Processing.* 2015. Vol. 24, no. 3. P. 862–872.
28. Karnaukhov V., Mozerov M. Fast non-local mean filter algorithm based on recursive calculation of similarity weights // *Journal of Communications Technology and Electronics.* 2018. Vol. 63, no. 12. P. 1475–1477.

## Impainting gradient reconstruction by non-local means filter

V.N. Karnaukhov, V.I. Kober, M.G. Mozerov, L.V. Zimina

Impainting is used both to improve the visual perception of an image, and in classical tasks of recognition and robotics in order to remove irrelevant information from an image. Modern methods of reconstruction drawing use neural networks. However, these approaches have disadvantages that do not allow the use of these algorithms in the practice of computer vision. The authors of this paper proposed in their recent paper an impainting method using a non-local mean filter (NLM) that outperforms other classical image processing methods in terms of PSNR. However, this method may have artifacts at the boundaries of the rendering zones, which distort visual perception. In this article, we propose to draw not the image, but the gradients of the restored image and only then restore the image itself, thus maintaining connectivity at the border of the rendering zones. This method overcomes most of the artifacts that arise when using the previously proposed method, and also surpasses it in terms of the recovery accuracy expressed by the PSNR criterion.

**KEYWORDS:** Impainting, non-local means filter, image restoration, image enhancement and contrasting.