

УДК:004.932.4

**Быстрая билатеральная фильтрация аэрофотоснимков на основе параллельной декомпозиции на пространственные фильтры**

**Беляева О.В.<sup>1\*</sup>, Пашенко О.Б.<sup>2\*\*</sup>, Филиппов М.В.<sup>1\*\*\*</sup>**

<sup>1</sup>*Московский государственный технический университет им. Н.Э. Баумана,  
ул. 2-я Бауманская, 5, Москва, 105005, Россия*

<sup>2</sup>*Российская самолетостроительная корпорация «МиГ»,  
1-й Боткинский проезд, 7, Москва, 125284, Россия*

*\*e-mail: [yacypress94@yandex.ru](mailto:yacypress94@yandex.ru)*

*\*\*e-mail: [alexandoleg@post.ru](mailto:alexandoleg@post.ru)*

*\*\*\*e-mail: [filippov.mike@mail.ru](mailto:filippov.mike@mail.ru)*

**Аннотация**

В работе рассматривается задача быстрой билатеральной фильтрация аэрофотоснимков, которая позволяет избавиться от малоразмерного шума и помех с сохранением резких границ объектов, что необходимо для первичной обработки кадров. Отсюда, акцент направлен на быстроту билатеральной фильтрации. Для решения задачи рассматривается метод билатеральной фильтрации на основе декомпозиции на пространственные фильтры, позволяющий фильтровать аэрофотоснимки на нескольких процессах одновременно. Для достижения быстродействия билатерального фильтра применяется его параллельная реализация на основе декомпозиции на рекурсивные пространственные фильтры Гаусса, что позволяет получать результаты в реальном времени.

**Ключевые слова:** аэрофотосъёмка, билатеральная фильтрация, рекурсивный фильтр Гаусса, распараллеливание.

## Введение

Введённый Tomasi и Manduchi [1] билатеральный фильтр, сохраняющий края, нашёл широкое применение во многих задачах по обработке изображений, например, фильтрация шума [2, 3, 4], редактирование текстуры [5] и тона [6], оценки оптического потока [7]. Билатеральная фильтрация также часто используется в качестве начального этапа обработки кадров, например, для задачи распознавания объектов [8], где необходимо отфильтровать несущественные детали и шумы при сохранении резких краев основного изображения.

Основным недостатком билатеральных фильтров являются большие вычислительные затраты. Поэтому в последнее время было опубликовано немало работ, посвященных реализации этих фильтров за сравнительно короткое время. Например, в работе [6] предложен метод билатеральной фильтрации, работающий за константное время, но предполагающий использование преобразования Фурье. В работах [9, 10] представлена реализация этого метода путём дискретизации значений интенсивности изображения и применении линейного фильтра для каждого дискретного значения. Далее по результатам пространственной фильтрации каждого значения интенсивности на выходе определяются PBFIC компоненты (Principle Bilateral Filtered Image Component). На следующем этапе применяется линейная интерполяция между двумя множествами PBFIC компонент.

В настоящей работе представлено усовершенствование метода, описанного в работе [10], за счет распараллеливания вычислительного процесса и применения рекурсивной фильтрации Гаусса [11]. Разработанный авторами алгоритм позволил существенно сократить время фильтрации при сохранении качества получаемых изображений.

## 1. Билатеральный фильтр

Билатеральная фильтрация (двунаправленная фильтрация) - это нелинейный и не итерационный процесс, комбинирующий пространственную (domain) и яркостную (range) фильтрацию. Таким образом, учитывается не только значения интенсивности близлежащих пикселей, но и их расстояние до текущего фильтруемого пикселя. Вклад близлежащих пикселей существенен по отношению к остальным. Двунаправленная фильтрация изображения  $f(x)$  в общем виде описывается выражением (1) [10]:

$$I^B(x) = k^{-1} \sum_{y \in N(x)} f_s(x, y) f_r(I(x), I(y)) * I(y), \quad (1)$$

$$\text{где } k = \sum_{y \in N(x)} f_s(x, y) f_r(I(x), I(y)); \quad (2)$$

$$f_s(x, y) = e^{-\frac{1}{2} \left( \frac{d(x-y)}{\sigma_s} \right)^2} \quad \text{- функция Гаусса по расстоянию между пикселями;}$$

$$f_r(I(x), I(y)) = e^{-\frac{1}{2} \left( \frac{I(x)-I(y)}{\sigma_r} \right)^2} \quad \text{- функция Гаусса по интенсивности.}$$

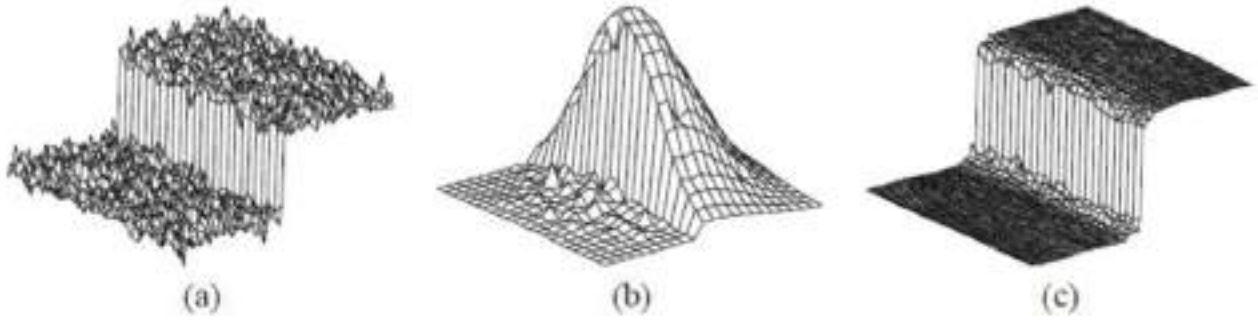


Рис. 1. (a) Шум Гаусса. (b) Комбинация двух функций  $f_s$  и  $f_r$  для зашумленного изображения (a). (c) Шаг (a) после билатеральной фильтрации.

Существенным недостатком фильтра является его скорость, так как вычисление изображения с окном размером  $M$  и пространственной коэффициентом  $\sigma_R$  необходимо  $O(M * \sigma_R^2)$ .

## 2. Декомпозиция билатерального фильтра на пространственные фильтры

На практике интенсивность пикселя  $I(x) \in \{0, \dots, N-1\}$  изображения – дискретна, где  $N$  – общее число значений оттенков серого.

При  $I(x) = k$  предполагается, что выражение (1) эквивалентно

$$I^B(x) = \frac{\sum_{y \in N(x)} f_s(x, y) f_r(k, I(y)) * I(y)}{\sum_{y \in N(x)} f_s(x, y) f_r(k, I(y))} \quad (3)$$

Для каждого пикселя  $y$  и для каждого значения интенсивности  $k \in \{0, \dots, N-1\}$  определяются выражения (4), (5):

$$W_k(y) = f_r(k, I(y)) \quad (4)$$

$$J_k(y) = W_k(y)I(y) \quad (5)$$

Отсюда билатеральный фильтр, декомпозированный на  $N$  множеств линейных фильтров можно представить в виде

$$I^B(x) = \frac{\sum_{y \in N(x)} f_s(x, y) J_k(y)}{\sum_{y \in N(x)} f_s(x, y) W_k(y)}, \quad (6)$$

где  $I^B(x) = J_{I(x)}^B, J_{I(x)}^B$  – компонента билатерального фильтра (PBFIC).

На практике используется только  $\hat{N}$  из  $N$  компонент PBFIC, где  $k \in \{L_0, \dots, L_{\hat{N}-1}\}$ , и значения фильтрации  $I^B(x)$  может быть линейно интерполировано от  $J_k^B$  до  $J_{k+1}^B$ :

$$I^B(x) = (L_{k+1} - I(x))J_k^B(x) + (I(x) - L_k)J_{k+1}^B(x) \quad (7)$$

Отметим что, ранговый фильтр  $f_r$  не ограничен, тогда он может быть представлен любой фильтрующей функцией. Однако существует ограничение в использовании малых значений  $\hat{N}$  для некоторых фильтров по причине их плохой аппроксимации. Рассматриваемый в работе фильтр Гаусса обладает такой характеристикой.

В соответствии с уравнениями (5), (6) в результате дискретизации только ранговая функция  $W_k(y)$  (4) чувствительна к шуму, по этой причине интенсивность пикселей входного изображения может не измениться в результате фильтрации.

Несмотря на ограничения метода, основанного на декомпозиции, он имеет следующие преимущества:

- 1) Основное вычисление метода приходится на вычисление  $\hat{N} * 2$  пространственных фильтров в соответствии с уравнением (6).
- 2) Возможность распараллеливания метода по компонентам PBFIC.

### 3. Параллельная реализация

Существенный вклад для достижения быстродействия билатерального фильтра вносит распараллеливание метода, основанного на декомпозиции на пространственные фильтры. В предыдущем разделе была описана реализация фильтрации путём линейной интерполяции между 2-мя соседними компонентами PBFIC. Трудоёмкость метода заключается в вычислении каждой компоненты, на которой производится две операции пространственной фильтрации. В работе [10] предлагают применять быстрые пространственные фильтры, например рекурсивный фильтр Гаусса, что вносит существенный вклад в быстродействие фильтра. Однако компоненты билатерального фильтра находятся независимо друг от друга, что позволяет распределить их вычисление по нескольким потокам, что позволит нам достигать вычислений в реальном времени.

Для достижения независимости вычисления компонент фильтра, необходимо было использовать дополнительный объём памяти. Для каждой компоненты, необходим буфер размером с входное изображение, для хранения результата пространственной фильтрации буфера  $J_k(y)$ , необходимого для интерполяции. Таким образом, для непараллельной реализации необходимо 3 буфера памяти размером  $N * M$  для матриц  $J_{k-1}(y), J_k(y), W_k(y)$ , где  $N$  и  $M$  – размеры входного изображения, и на каждой итерации производить линейную интерполяцию пикселей

попадающих в диапазон текущей компоненты. Для параллельной реализации необходимо знать сразу все значения  $J_k(y)$  для каждой компоненты, таким образом, общий размер памяти в данном случае составит: count(компонент PBFIC) + 1 буферов размером  $N \times M$  для матриц  $J_0(y), \dots, J_p(y), W_k(y)$ , где  $p$  – количество компонент PBFIC. Таким образом, существенное преимущество по времени происходит за счёт потребления дополнительной памяти.

#### 4. Пространственный рекурсивный фильтр Гаусса

Согласно выражению (6) билатеральный фильтр с пространственным и ранговым ядрами декомпозируется на 2 множества пространственных фильтров  $J_k(y)$  и  $W_k(y)$ . Таким образом, вычислительная сложность метода зависит от вычисления пространственных фильтров. Для повышения скорости работы билатерального фильтра, было решено оптимизировать вычисление пространственного фильтра Гаусса с помощью использования рекурсивного метода, изложенного в работе [11], путём аппроксимации преобразования Фурье ядра Гаусса, вычисляющего результаты за константное время работы.

$$g(t, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{t^2}{2\sigma^2}} \quad (8)$$

$$G(\omega; \sigma) = \int_{-\infty}^{+\infty} dt e^{-i\omega t} g(t; \sigma) = e^{-\omega^2 \sigma^2 / 2} \quad (9)$$

Бесконечная импульсная характеристика (БИХ) приближена к фильтрации Гаусса и не зависит от его ширины выраженной  $\sigma$ . БИХ Гаусса может быть представлена рекурсивной последовательностью:

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] - a_1 y[n-1] - a_2 y[n-2] \quad (10)$$

где  $b_0, b_1, b_2, a_1, a_2$  – коэффициенты фильтра.

Разностное уравнение (8) эквивалентно уравнению:

$$y[n] = y[n-1] + \frac{x[n+M] - x[n-M-1]}{2M+1} \quad (11)$$

Решение рекурсивного разностного уравнения не является точным решением фильтрации Гаусса, так как не достаточно использования только 5-ти параметров для формирования импульсной характеристики системы, повторяющей функцию Гаусса. Кроме того, импульсный отклик системы согласно уравнению (8) – односторонний, следовательно - не симметричен, что противоречит функции Гаусса. По этой причине симметричная фильтрация БИХ достигается путём решения 2-х систем 2-го порядка, описываемые выражениями  $H^-$  и  $H^+$ , представляющих каузальную (14) и некаузальную (15) системы в z-преобразованиях соответственно.

Приняв  $z = e^{i\omega_0} = e^{i(\nu_0 + i\mu_0)} = e^{-\mu_0} e^{i\nu_0}$ , частотная характеристика системы выглядит следующим образом:

$$H(\omega; \sigma) \approx H(\omega\sigma; \sigma_0) = \frac{1}{1 - e^{i(\omega_0 - \omega\sigma)}} \bar{H}(\omega\sigma; \sigma_0) \quad (12)$$

В работе [10] было предложено использовать для построения рекурсивного фильтра Гаусса сумму каузальной  $H^-$  и некаузальной  $H^+$  систем



$H(z) = H^+(z) + H^-(z)$ . Таким образом, данные фильтры 4-го порядка  $D$  определяются следующим образом:

$$H^+(z) = \frac{b_0^+ + b_1^+ z^{-1} + b_2^+ z^{-2} + b_3^+ z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}} \quad (14)$$

$$H^-(z) = \frac{b_1^- z^1 + b_2^- z^2 + b_3^- z^3 + b_4^- z^4}{1 + a_1 z^1 + a_2 z^2 + a_3 z^3 + a_4 z^4} \quad (15)$$

Коэффициенты  $a_1, a_2, a_3, a_4$  одинаковы для 2-х систем  $H^+(z)$  и  $H^-(z)$  зависят только от положения фильтра. Коэффициенты  $b_1^+, b_2^+, b_3^+, b_4^+$  в системе  $H^+(z)$  зависят от расположения нулей фильтра. Остальные коэффициенты  $b_1^-, b_2^-, b_3^-, b_4^-$  выражаются из остальных, так как структура фильтра  $H(z)$  – симметрична .

$$H(z) = H^+(z) + H^-(z) = H(z^{-1}) \quad (16)$$

$$b_1^- = b_1^+ + b_0^+ a_1$$

$$b_2^- = b_2^+ + b_0^+ a_2$$

$$b_3^- = b_3^+ + b_0^+ a_3$$

$$b_4^- = b_4^+ + b_0^+ a_4$$

## 5. Результаты

В этом разделе представлены экспериментальные результаты работы метода билатерального фильтра, основанного на декомпозиции на пространственные фильтры. Основной акцент делается на быстродействие метода, достигаемое за счёт

распараллеливания метода и применения рекурсивного пространственного фильтра Гаусса для матриц  $J_k(y)$  и  $W_k(y)$ .

Экспериментальные испытания проводились на платформе со следующими характеристиками:

1. Процессор Intel Core i7 1.9 ГГц;
2. 64-х разрядная ОС Windows 8.1;
3. 4 Гб оперативной памяти.

На рисунке 2 (а), (б) представлены результаты фильтрации простым билатеральным фильтром и билатеральным фильтром на основе декомпозиции на пространственные фильтры с параллельной реализацией. Экспериментальные результаты были получены с использованием 4-х потоков.



(а) Исходное изображение



(б) Простой билатеральный фильтр



(в) Быстрый билатеральный фильтр

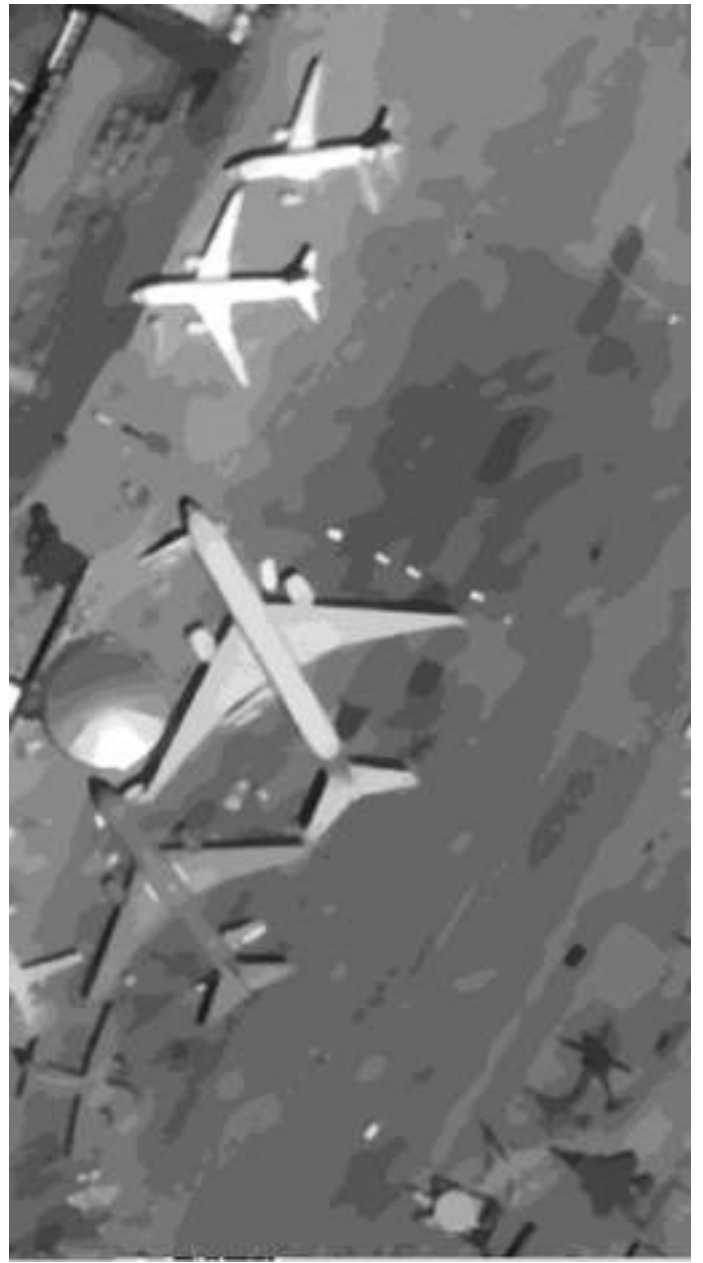
(г) Быстрый билатеральный фильтр

Рисунок 2. (а) Исходное изображение. (б) Результат фильтрации простым билатеральным фильтром при  $\sigma_R^2 = 5$ ,  $\sigma_S^2 = 5$ . Качество кадра 29.3 SNR. (в) Результат фильтрации быстрым билатеральным фильтром на основе декомпозиции на пространственные фильтры с использованием распараллеливания при  $\sigma_R = 0.1$ ,  $\sigma_S = 10$ , при фильтрации использовалось 10 компонент (PBFIC). Качество кадра 25 SNR. (г) Результат фильтрации быстрым билатеральным фильтром при  $\sigma_R = 0.2$ ,  $\sigma_S = 1, 5$  PBFIC. Качество кадра 21 SNR.

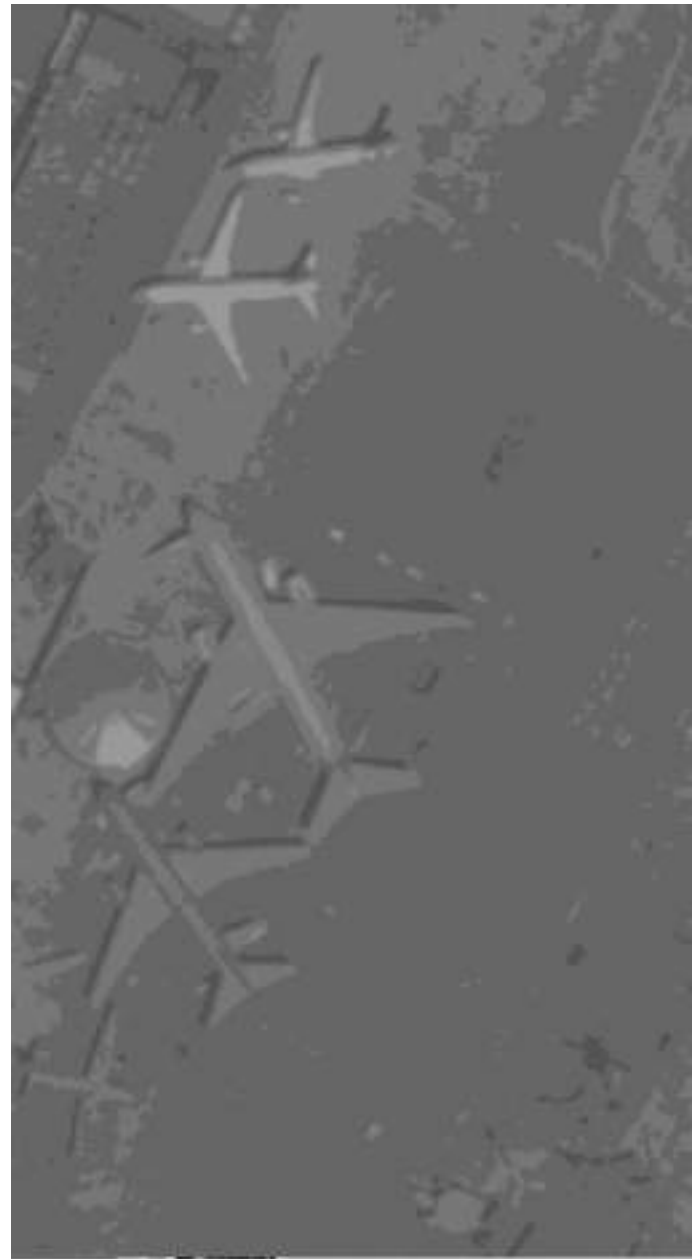
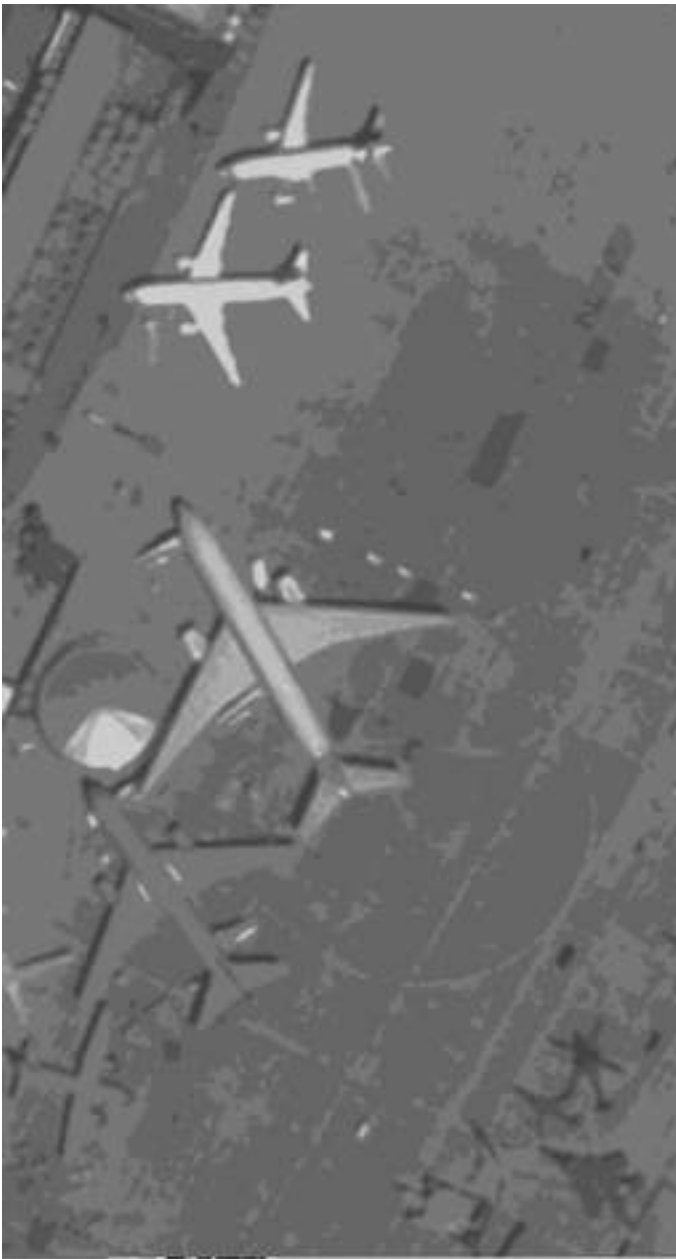
Основное преимущество рассматриваемого билатерального фильтра на основе декомпозиции является его быстрое действие. Результат фильтрации нашим методом с учётом распараллеливания 2(а) размером 618x612, представленный на рисунке 2(б) получен за 0.86 секунд, 2(г) – за 0.41 секунду, в то время как простая билатеральная фильтрация исходного изображения 2(а) выполнялась за 61 секунду.



(a) Исходное изображение



(б) Простой билатеральный фильтр



(в) Быстрый билатеральный фильтр

(г) Быстрый билатеральный фильтр

Рисунок 3. (а) Исходное изображение. (б) Результат фильтрации простым билатеральным фильтром при  $\sigma_R^2=5$ ,  $\sigma_S^2=5$ . Качество кадра 32.7 SNR. (в) Результат фильтрации быстрым билатеральным фильтром на основе декомпозиции на пространственные фильтры с использованием распараллеливания при  $\sigma_R = 0.1$ ,  $\sigma_S = 10$ , при фильтрации использовалось 9 компонент (PBFIC). Качество кадра 26.5 SNR.

(г)  $\sigma_R = 0.15$ ,  $\sigma_S = 3$ , при фильтрации использовалось 6 компонент (PBFIC). Качество кадра 20.3 SNR.

Результат фильтрации с учётом распараллеливания на изображении 3(а) размером 299x554, представленный на рисунке 3(в) получен за 0.6 секунд, 3(г) за 0.5 секунды, в то время как простая билатеральная фильтрация исходного изображения 3(а) выполнялась за 56.9 секунд.

По результатам видно, что рекурсивный фильтр Гаусса, использованный для повышения быстродействия фильтра, повышает контрастность изображения. В целом, качество восстановления быстрым билатеральным фильтром ниже, но в задачах требующим обработки кадров в реальном времени, этот недостаток опускается по сравнению с его преимуществом – высокой скоростью работы.

### **Выводы**

В работе был представлен быстрый билатеральный фильтр на основе декомпозиции на простые пространственные фильтры с использованием распараллеливания по компонентам (PBFIC). Экспериментальные результаты показывают очевидное преимущество метода по быстродействию за счёт дополнительного использования памяти. Несмотря на это, достигнуто время работы билатерального фильтра в реальном времени, что является существенным преимуществом для использования фильтра в качестве предварительной обработки кадров.

## Библиографический список

1. Tomasi C., Manduchi R. Bilateral Filtering for Grey and Color Images // Proceedings of the 1998 IEEE International Conference on Computer Vision. Bombay. India. 1998, pp. 839–846.
2. Buades, A., Coll, B., Morel, J.M. A review of image denoising algorithms, with a new one // Multiscale Modeling and Simulation. 2005. Vol. 4, pp. 490–530.
3. Yang, Q., Yang, R., Davis J., Nist'er D. Spatial-depth super resolution for range images // CVPR, IEEE Computer Society Conference on Computer Vision and Pattern. 2007. no. 15, pp 1-8.
4. Wong, W.C.K., Chung, A.C.S., Yu, S.C.H.: Trilateral filtering for biomedical images // In Proceedings of the IEEE International Symposium on Biomedical Imaging. 2004, pp. 820-823.
5. Oh, B.M., Chen, M., Dorsey, J., Durand, F. Imagebased modeling and photo editing // Proceedings of the 28th annual conference on Computer graphics and interactive techniques Siggraph, 2001, pp. 433-442.
6. Durand, F., Dorsey, J. Fast bilateral filtering for the display of high-dynamic-range images // Proceedings of the ACM SIGGRAPH conference. 2002, pp.253-263.
7. Xiao, J., Cheng, H., Sawhney, H., Rao, C., Isnardi, M. Bilateral filtering-based optical flow estimation with occlusion detection // Computer Vision–ECCV 2006. Computer Vision–ECCV. 2006. Vol. 21, pp. 211-224.

8. Князь В. В., Бусурин В. И. Автоматическое обнаружение препятствий на взлётно-посадочной полосе средствами технического зрения // Труды МАИ. 2015. № 81. URL: <http://www.mai.ru/science/trudy/published.php?ID=57839>
9. Porikli F. Constant time  $O(1)$  bilateral filtering // IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska. USA. 2008. pp. 1–8.
10. Yang Q., Tan K.-H., Ahuja N. Real-time  $O(1)$  bilateral filtering // In Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, (CVPR '09), June 2009, pp. 557–564.
11. Deriche R. Recursively implementing the gaussian and its derivatives // in Proceedings of IEEE International Conference on Image Processing, (ICIP '92), 1992, pp. 263–267.