

УДК: 004.932.4

## ВОССТАНОВЛЕНИЕ ФОНА В ВИДЕО С ИСПОЛЬЗОВАНИЕМ НЕПАРАМЕТРИЧЕСКОЙ МОДЕЛИ ДВИЖЕНИЯ И ПОКАДРОВОГО УТОЧНЕНИЯ

*Боков А.А., аспирант лаборатории компьютерной графики и мультимедиа МГУ им. М. В. Ломоносова, e-mail: abokov@graphics.cs.msu.ru;*

*Ватолин Д.С., к.ф.-м.н., с.н.с. лаборатории компьютерной графики и мультимедиа МГУ им. М. В. Ломоносова, e-mail: dmitriy@graphics.cs.msu.ru.*

### VIDEO BACKGROUND RECONSTRUCTION BY NON-PARAMETRIC MOTION COMPLETION AND FRAME-BY-FRAME REFINEMENT

*Bokov A.A., Vatolin D.S.*

*This paper introduces a novel video background reconstruction algorithm that combines the generality and the level of quality provided by the state-of-the-art approaches with much lower computational complexity, which is possible owing to per-frame processing instead of global spatio-temporal optimization. We achieve this result by using a general non-parametric method for motion completion and a frame-by-frame refinement step to alleviate the effect of error accumulation. The quality of the proposed approach is confirmed by the conducted objective evaluation and comparison with existing background reconstruction algorithms.*

**Key words:** background reconstruction, motion completion, video processing.

**Ключевые слова:** восстановление фона, модель движения, покaдровая обработка, непараметрическая, вычислительная сложность.

#### Введение

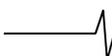
Восстановление фона является актуальной задачей интеллектуальной обработки видеопоследовательностей и имеет множество различных приложений. Оно применяется в киноиндустрии как в современных постановках для удаления нежелательных объектов из сцены, так и для восстановления архивных видеозаписей, содержащих дефекты пленки. Все большую популярность на сегодняшний день получает такой способ производства стереоскопических фильмов как 2D-3D конвертация, один из этапов которого заключается в восстановлении фона за объектами переднего плана, также известный как заполнение областей открытия. Более 60 % стереоскопических фильмов, вышедших в 2015-2016 годах в мировой прокат, были произведены именно путем конвертации из моноскопического формата. Также стоит упомянуть про такие приложения как маскирование ошибок видеокодека, вызванных потерей пакетов при передаче по сети [1], и заполнение неизвестных областей, возникающих при стабилизации видео с сохранением исходного разрешения [2].

Задача восстановления фона в видео, как правило, ставится следующим образом: на вход алгоритма помимо исходной видеопоследовательности подается бинарная маска неизвестной области  $\Omega$ , где требуется восстановить фон на базе известной части входного видео  $V/\Omega$ . Результатом работы алгоритма является восстановленный фон в области  $\Omega$ , который оценивается по степени правдоподобности с точки зрения стороннего наблюдателя (обработанная видеопоследовательность не должна содержать видимых следов модификации). Данная задача достаточно широко исследована в

*Представлен новый алгоритм восстановления фона в видео, который сопоставим с современными аналогами по уровню качества и ограничениям, накладываемым на входные данные, и при этом обладает существенно более низкой вычислительной сложностью за счет использования покaдровой обработки вместо решения глобальной задачи оптимизации по всему пространственно-временному объему. Данный результат обеспечивается использованием общей непараметрической модели движения фона и покaдрового уточнения результата, что предотвращает возможное накопление ошибки. Качество работы предложенного алгоритма подтверждается проведенным объективным сравнением с существующими аналогами.*

литературе, однако до недавнего времени автоматические методы накладывали существенные ограничения на входную видеопоследовательность. Целый класс алгоритмов, известный как алгоритмы вычитания фона [3], опирается на предположение о статичности камеры для проведения восстановления. Многие предложенные алгоритмы используют простые параметрические модели для компенсации движения камеры, такие как аффинное [4], проективное [5] и кусочно-проективное [6] преобразования. Также многие алгоритмы [6, 7] делают предположение о том, что восстанавливаемые фрагменты фона должны быть видны хотя бы в одном кадре входного видео, что не всегда выполняется на практике. Современные подходы [8] во многом лишены вышеперечисленных недостатков, однако их применение существенно затрудняется высокой вычислительной сложностью и потреблением памяти, связанным с постановкой задачи восстановления фона как задачи оптимизации определенного функционала одновременно по всему пространственно-временному объему.

В данной работе представлен новый алгоритм восстановления фона в видео, основанный на использовании непараметрической модели движения и покaдровой оптимизации для уточнения результатов восстановле-



ния. Предложенный подход обладает значительно меньшей вычислительной сложностью, чем современные аналоги [4, 8] и позволяет обрабатывать видеопоследовательности с произвольным движением камеры и восстанавливать фрагменты фона, не видимые ни в одном из кадров входного видео. Качество работы предложенного алгоритма экспериментально подтверждено проведенным объективным сравнением с шестью аналогами, включая коммерческие инструменты для восстановления фона, с использованием тестовой выборки и методики оценки качества, описанной в [9]. Также проводится качественное сравнение предложенного алгоритма с современным аналогом [8] путем приведения результатов работы на репрезентативных кадрах.

### Обзор области исследований

Одним из классических подходов к восстановлению фона в видео является использование трехмерных пространственно-временных блоков для оценки локального сходства восстановленного фона с известной частью входной видеопоследовательности  $V / \Omega$ . В работе [10] задача формулируется как задача глобальной оптимизации по всему пространственно-временному объему, в ходе которой непосредственно минимизируется сумма расстояний от каждого трехмерного блока неизвестной области до наиболее схожего блока в известной части входного видео. Для нахождения наиболее схожего трехмерного блока в видео используется полный перебор и его простые оптимизации, что ограничивает область применимости метода до коротких последовательностей низкого разрешения (авторы приводят результаты на видео разрешения  $360 \times 288$ ). В работе [4] был предложен способ значительного ускорения этапа поиска наиболее схожего трехмерного блока во входном видео путем использования алгоритма PatchMatch [12]. Авторами была модифицирована метрика расстояния между трехмерными блоками путем включения текстурных признаков, что позволило улучшить качество восстановления динамических текстур в видео, таких как волны, огонь, толпа людей. Однако, как было продемонстрировано в недавней работе [8], подходы, основанные на использовании трехмерных блоков, обладают существенными ограничениями при обработке видеопоследовательностей с произвольным движением камеры.

Некоторые алгоритмы накладывают явные ограничения на допустимый характер движения камеры и/или объектов во входном видео. В работе [13] авторы допускают лишь плоскопараллельное движение камеры во входной сцене. Авторы работы [14, 15] исходят из предположения, что входное видео содержит движущиеся объекты переднего плана и статичный фон. В [5] для компенсации движения камеры используется глобальное проективное преобразование, которое неспособно моделировать движение более чем одной плоскости в сцене при произвольном движении камеры. В работе [6] модель движения расширяется до кусочно-проективного преобразования (движение камеры компенсируется путем нахождения нескольких плоскостей в сцене и соответствующих им проективных преобразований). Авторы работы [6] также предлагают обобщение стандартного

метода бесшовного монтажа на случай восстановления фона в видео для компенсации возможных изменений освещенности. Однако данный алгоритм основан на попарном выравнивании всех входных кадров, что приводит к квадратичной зависимости времени работы алгоритма от количества кадров во входном видео. Авторы отмечают, что на обработку видео разрешения  $960 \times 720$  у предложенного алгоритма уходит до 4 часов при работе на сервере с 64 логическими процессорами. В работе [16] используется схожий подход, но радиус поиска ограничивается соседними кадрами для снижения вычислительной сложности (авторы предлагают использовать скользящее окно, состоящее из 21 кадра). Несмотря на это, предложенный алгоритм сохраняет сравнительно высокую вычислительную сложность, которая затрудняет его применение на практике (авторы заявляют скорость работы в 1,5 минуты на кадр разрешения  $1440 \times 1056$ ).

Наиболее схожими с предлагаемым в данной статье алгоритмом являются алгоритмы, основанные на непараметрических методах восстановления поля векторов движения в неизвестной области. В некоторых работах [11, 17, 18] восстановление поля векторов движения осуществляется путем копирования векторов из известной области с использованием трехмерных блоков для оценки локальной схожести поля векторов. Основным ограничением такого подхода является невозможность корректного восстановления поля векторов движения в неизвестной области простым копированием для некоторых типов движения камеры и изменений характеристик оптической системы объектива (например, масштабирование или наезд камеры). Авторы работы [19] применяют схожую технику с использованием трехмерных блоков, однако дополнительно выполняют итеративное уточнение поля векторов движения в неизвестной области для повышения качества. В работе [20] авторы сначала строят поле векторов движения в известной области и затем производят его покадровую интерполяцию в неизвестную область с использованием стандартного метода, основанного на диффузии. Далее непосредственно для восстановления фона авторы модифицируют метрику расстояния между двумерными блоками с учетом восстановленного поля векторов движения для поддержания временной стабильности. Поиск наиболее схожих двумерных блоков по всему входному видео, аналогично ранее рассмотренным алгоритмам, приводит к высокой вычислительной сложности (авторы сообщают, что предложенный алгоритм выполнил обработку видео разрешения  $1280 \times 720$  продолжительностью 82 кадра за 3,3 часа). В другой работе [7] формулируется задача глобальной оптимизации для совместного нахождения восстановленного фона и соответствующего ему поля векторов движения. Авторы описывают итеративный алгоритм решения данной задачи, основанный на чередовании вычисления поля векторов движения во всей видеопоследовательности на базе текущего восстановленного фона и вычисления восстановленного фона на базе текущего поля векторов движения в неизвестной области  $\Omega$ . Данный подход был далее развит в работе [8], где авторы демонстрируют высокое качество

восстановления фона на широком классе видеопоследовательностей с произвольным движением камеры и наличием фрагментов фона, не видимых ни в одном из кадров входного видео. Однако вследствие формулировки задачи как глобальной оптимизации по всему пространственно-временному объему, данный алгоритм сохраняет проблему подавляющего большинства существующих алгоритмов восстановления фона, которая заключается в чрезмерно высокой вычислительной сложности (предложенному авторами алгоритму требуется порядка 3 часов на обработку видео 854×480 продолжительностью 90 кадров).

Среди алгоритмов восстановления фона низкой вычислительной сложности стоит особо отметить работу [21], где предлагается специализированный алгоритм для восстановления фона в областях открытия, возникающих при генерации 3D-видео, основанный на локальной экстраполяции поля векторов движения. Однако данный алгоритм обладает рядом ограничений. Во-первых, использование локального копирования векторов движения в неизвестную область позволяет корректно восстанавливать движение фона лишь при плоско параллельном движении камеры и неспособно корректно обрабатывать сцены с существенным масштабированием или поворотом. Предложенный авторами алгоритм предполагает многократное копирование восстановленного фона вдоль восстановленных векторов движения с использованием бикубической интерполяции, что приводит к накоплению ошибки и размытию восстановленного фона, если он находится в рамках неизвестной области в течение длительного времени. Авторы работы [22] предлагают алгоритм восстановления фона, способный работать в режиме реального времени, однако данный алгоритм состоит в использовании глобального проективного преобразования для покадрового распространения результатов пространственного восстановления фона. То есть, предложенный авторами подход никак не использует фрагменты восстанавливаемого фона, видимые в других кадрах входной видеопоследовательности.

Таким образом, существующие алгоритмы восстановления фона либо накладывают ограничения на допустимые входные видеопоследовательности, либо обладают чрезмерно высокой вычислительной сложностью, что затрудняет использование соответствующих алгоритмов на практике. В то же время, предлагаемый в данной работе алгоритм не накладывает существенных ограничений на исходную видеопоследовательность, позволяя обрабатывать видео с произвольным движением камеры, благодаря использованию непараметрической модели движения; также он обладает более низкой вычислительной сложностью в сравнении с аналогами за счет применения покадровой оптимизации вместо преимущественно используемой в современных алгоритмах глобальной оптимизации по всему пространственно-временному объему.

### Предлагаемый алгоритм восстановления фона

Как было отмечено ранее, отличительной особенностью предлагаемого алгоритма является покадровая

обработка входной видеопоследовательности. Алгоритм выполняет три прохода по исходной видеопоследовательности (в прямом направлении, в обратном, и снова в прямом), выполняя обработку очередного кадра с использованием результатов обработки предыдущего. В каждом из проходов неизвестная область в текущем кадре может быть восстановлена путем копирования фрагментов фона из любого из уже обработанных в данном проходе кадров (темпоральное восстановление). На отдельных кадрах также осуществляется восстановление фона путем копирования фрагментов из известной области того же самого кадра (пространственное восстановление). После каждого из проходов весь восстановленный фон фиксируется и используется как известный фон в следующем проходе. Обработка одного кадра в предложенном алгоритме включает в себя следующий набор шагов:

Восстановление поля векторов движения фона  $O^{t \rightarrow t_0}$  внутри маски неизвестной области  $\Omega^t$  от текущего кадра  $I^t$  к предыдущему обработанному кадру  $I^{t_0}$  (см. далее (1));

Построение поля векторов  $V^t : \Omega^t \rightarrow R^2 \times N$ , которое определяет отображение из неизвестной области на текущем кадре с номером  $t$  в известные области уже обработанных в данном проходе кадров. Для каждого пиксела в  $\Omega^t$  данное поле векторов содержит либо носительный пространственный сдвиг и индекс исходного кадра, содержащего известный фон, либо особое значение  $V_{undef}$ , означающее, что в данном пикселе отображение в известный фон не определено. Данное поле векторов вычисляется с использованием результата предыдущего кадра  $V^{t_0}$  для построения начального приближения (см. далее (4));

Применение модифицированного алгоритма бесшовного монтажа для интеграции известных фрагментов фона, определенных полем векторов  $V^t$ , в текущий кадр с учетом возможных изменений условий освещенности сцены (см. далее (5));

Применение алгоритма пространственного восстановления фона на первом и последнем кадрах для обработки фрагментов неизвестной области, которые остаются скрытыми на протяжении всей входной видеопоследовательности.

Таким образом, предлагаемый алгоритм позволяет полностью использовать известный фон на всех кадрах входной видеопоследовательности, наряду с пространственным восстановлением на первом и последнем кадрах для фрагментов фона, не видимых ни в одном из кадров входного видео. Далее основные шаги алгоритма будут рассмотрены более подробно.

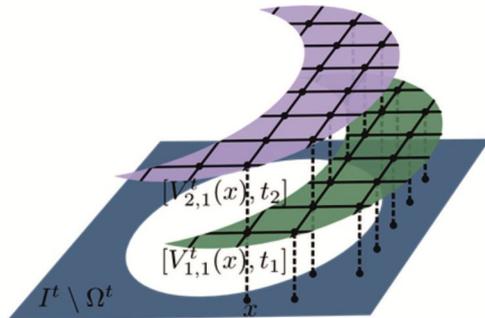
### Восстановление поля векторов движения в неизвестной области

Существующие методы восстановления поля векторов движения либо используют стандартные алгоритмы построения оптического потока для известной области и отдельно формулируют задачу восстановления поля векторов движения в неизвестной области по известной [2, 17, 23], либо формулируют глобальную задачу опти-

мизации по всему видео для совместного нахождения цвета восстановленного фона и его движения [7, 8]. Отличительной особенностью предлагаемого в данной статье подхода является формулирование единой задачи оптимизации для совместного нахождения поля векторов движения в известной области и его гладкой интерполяции в неизвестной области с сохранением границ. Также, предложенный подход отличается от существующих алгоритмов построения оптического потока совмещением блочной оценки ошибки сопоставления и ограничений на гладкость поля векторов, характерных для вариационных алгоритмов построения оптического потока. А именно, поле векторов движения  $O^{t \rightarrow t_0}$  от  $t$ -ого кадра до  $t_0$ -ого вычисляется в узлах однородной сетки с шагом в  $s$  пикселей с использованием блоков  $P(x, y)$  размера  $s \times s$  для оценки ошибки сопоставления (в авторской реализации  $s = 8$  пикселей) путем решения следующей оптимизационной задачи:

$$O^{t \rightarrow t_0} = \arg \min_O \sum_{(x, y)} \Phi \left( \sum_{\substack{p \in P(x, y) \\ p \notin \Omega^t \\ p+O(x, y) \in \Omega^0}} w_p (I^t(p) - I^{t_0}(p + O(x, y)) - M(x, y))^2 + \alpha \Phi(\|O(x+1, y) - O(x, y)\|^2 + \|O(x, y+1) - O(x, y)\|^2), \right. \\ \left. M(x, y) = \frac{\sum_p w_p (I^t(p) - I^{t_0}(p + O(x, y)))}{\sum_p w_p} \right) \quad (1)$$

Здесь:  $\Phi(a^2) = \sqrt{a^2 + 0,001}$  – робастная функция потерь, позволяющая сохранить разрывы в поле векторов движения,  $w_p = 1 / (1 + \|\nabla I^t(p)\|^2)$  – нормировочные веса, предотвращающие чрезмерное влияние областей с высокими значениями модуля градиента;  $M(x, y)$  представляет собой нормировку на средние яркости блоков, что повышает устойчивость к изменениям освещения между соседними кадрами. Параметр  $\alpha$  определяет степень гладкости получаемых полей векторов движения. В целях уменьшения вычислительной сложности при решении данной оптимизационной задачи



учитывается лишь яркостная компонента входной видеопоследовательности.

Аналогично существующим алгоритмам построения оптического потока [31], оптимизация данного функционала производится в рамках Гауссовой пирамиды с использованием разложения изображения в ряд Тейлора в предположении, что приращение вектора движения достаточно мало. Каждый следующий уровень пирамиды меньше предыдущего в 2 раза по каждой из осей; количество уровней пирамиды определяется разрешением исходного видео таким образом, чтобы нижний уровень пирамиды был последним, в который помещается хотя бы один блок  $s \times s$  целиком. На каждом уровне пирамиды производится  $N_{outer}$  внешних итераций с разложением в ряд Тейлора до первого порядка включительно вокруг текущей аппроксимации вектора движения в каждом блоке. Из-за использования неквадратичной функции потерь, на каждую внешнюю итерацию также приходится  $N_{fp}$  итераций, в каждой из которых производится  $N_{SOR}$  итераций метода релаксации для решения соответствующей системы линейных уравнений (в авторской реализации  $N_{outer} = 10$ ,  $N_{fp} = 5$ ,  $N_{SOR} = 25$ ). Финальное попиксельное поле векторов движения в неизвестной области получается путем билинейной интерполяции значений на разреженной сетке.

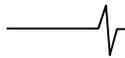
#### Построение поля векторов до исходных кадров

Первое приближение поля векторов  $V^t: \Omega^t \rightarrow R^2 \times N$ , определяющего пространственный сдвиг и номер исходного кадра с известным фоном для каждого пиксела неизвестной области  $\Omega^t$ , может быть построено путем простого накопления векторов движения в неизвестной области, которые были вычислены на предыдущем шаге. А именно, первое приближение  $V^t$  строится путем прибавления поля векторов движения  $O^{t \rightarrow t_0}$  к результату предыдущего кадра  $V^{t_0}$ :

$$V^t(x) = \begin{cases} [O^{t \rightarrow t_0}(x), t_0], & x_0 \notin \Omega^{t_0} \\ [O^{t \rightarrow t_0}(x) + V_1^{t_0}(x_0), V_2^{t_0}(x_0)], & V^{t_0}(x_0) \neq V_{undef} \\ V_{undef}, & иначе. \end{cases} \quad (2)$$

- – известный фон в текущем кадре
- – отображение в исходный кадр  $t_1$
- – отображение в исходный кадр  $t_2$
- ⋯ – ограничения на выравнивание фрагментов фона с разных исходных кадров
- ⋯ – ограничения на гладкость отображений в каждый исходный кадр

Рис. 1. Схема, иллюстрирующая расширенное определение поля векторов  $V^t$ , допускающее наличие нескольких отображений в различные исходные кадры в одном пикселе неизвестной области. На этапе уточнения поля векторов совместно оптимизируется выравнивание между фрагментами фона с разных кадров и гладкость отображений в каждый из них



Здесь:  $x_0 = x + O^{t \rightarrow t_0}(x)$ ,  $V_1^{t_0}(x_0)$  – первый элемент  $V^{t_0}(x)$  (значение пространственного сдвига) и  $V_2^{t_0}(x_0)$  – второй элемент (номер исходного кадра). Имея отображение в известный фон на другом кадре, определенное  $V^t(x)$ , можно восстановить фон в пикселе  $x$  простым копированием с билинейной интерполяцией. Однако экспериментально было установлено, что использование лишь восстановленных векторов движения в неизвестной области может приводить к накоплению ошибки. Для корректировки накопленной ошибки в данной работе предлагается алгоритм покадрового уточнения поля векторов  $V^t$ , основанный на оптимизации функционала, учитывающего как гладкость результирующего поля векторов, так и выравнивание между фрагментами фона, восстановленными с различных исходных кадров.

Для описания предложенного алгоритма уточнения поля векторов  $V^t$  введем ряд дополнительных обозначений. Во-первых, расширим определение  $V^t$ , чтобы оно допускало наличие нескольких отображений в известный фон на различных исходных кадрах в каждом пикселе  $x \in \Omega^t$  (см. иллюстрацию на рис. 1). Точнее, будем предполагать, что пиксел  $x$  имеет  $F^t(x)$  отображений в различные исходные кадры:

$$V^t(x) = \left\{ [V_{1,1}^t(x), V_{1,2}^t(x)], \dots, [V_{F^t(x),1}^t(x), V_{F^t(x),2}^t(x)] \right\}, \quad (3)$$

где  $V_{i,1}^t(x), V_{i,2}^t(x)$  – это, соответственно, пространственный сдвиг и номер исходного кадра  $i$ -ого отображения. Следовательно,  $V_{1,2}^t(x), \dots, V_{F^t(x),2}^t(x)$  образует множество различных номеров исходных кадров, в которые пиксел  $x$  имеет отображения. Также расширим область определения  $V^t$  до  $\Omega_d^t$  – результата морфологического расширения  $\Omega^t$  на  $d$  пикселей. С помощью данного обобщения мы можем ввести перекрытия между отображениями из неизвестной области в различные исходные кадры, что позволяет учитывать выравнивание между ними в процессе уточнения, причем ширина такого перекрытия становится параметром алгоритма  $d$  (в авторской реализации  $d = 6$  пикселей). Таким образом, каждый пиксел в расширенной неизвестной области  $\Omega_d^t$  содержит не более  $d + 1$  отображений в различные исходные кадры. Поэтому описанную выше процедуру построения первого приближения  $V^t$  по результату предыдущего кадра  $V^{t_0}$  можно обобщить на расширенное определение  $V^t$ , допускающее перекрытия, с сохранением временной сложности пропорциональной количеству пикселей в  $\Omega_d^t$ , вне зависимости от количества кадров во входной видеопоследовательности.

Сам алгоритм уточнения формулируется как задача оптимизации функционала, содержащего слагаемое  $E_D(V^t)$ , которое накладывает ограничение на выравнивание между фрагментами фона, восстановленными из различных исходных кадров, и слагаемое  $E_S(V^t)$ ,

которое налагает штраф на разрывы в полях векторов, соответствующих отображениям в каждый из исходных кадров:

$$V^t = \arg \min_{V^t} (E_D(V^t) + \lambda E_S(V^t)), \quad (4)$$

$$E_D(V^t) = \sum_{x \in \Omega_d^t \setminus \Omega^t} \sum_{i=1}^{F^t(x)} \Phi((I^{V_{i,2}^t(x)}(x + V_{i,1}^t(x)) - I^t(x))^2) +$$

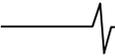
$$+ \sum_{x \in \Omega^t} \sum_{i=1}^{F^t(x)-1} \Phi((I^{V_{i+1,2}^t(x)}(x + V_{i+1,1}^t(x)) - I^{V_{i,2}^t(x)}(x + V_{i,1}^t(x)))^2),$$

$$E_S(V^t) = \sum_{(x_1, x_2) \in e(\Omega_d^t)} \sum_{\substack{1 \leq i_1 \leq F^t(x_1) \\ 1 \leq i_2 \leq F^t(x_2) \\ V_{i_1,2}^t(x_1) = V_{i_2,2}^t(x_2)}} \|V_{i_1,1}^t(x_1) - V_{i_2,1}^t(x_2)\|^2.$$

Здесь:  $e(\Omega_d^t)$  – множество всевозможных пар соседних пикселей  $(x_1, x_2)$ , принадлежащих  $\Omega_d^t$ . Первое слагаемое в  $E_D(V^t)$  отвечает за выравнивание с известным фоном в текущем кадре, второе слагаемое отвечает за выравнивание фрагментов фона, восстановленных из различных исходных кадров, между собой. Оптимизация данного функционала проводится с помощью разложения в ряд Тейлора, аналогично оптимизации функционала (1). Для снижения вычислительной сложности также используется лишь яркостная компонента исходного видео. Однако здесь оптимизация происходит только на оригинальном разрешении, и разложение в ряд Тейлора применяется лишь один раз, так как использование восстановленного поля векторов движения фона уже предоставляет хорошее начальное приближение. Возникающая система линейных уравнений решается методом сопряженных градиентов с числом итераций  $N_{CG} = 50$  в авторской реализации. Благодаря использованию результата предыдущего кадра для построения первого приближения поля векторов, даже такое малое число итераций позволяет получить решение приемлемого качества. В зависимости от структуры  $V^t$  количество неизвестных в СЛАУ может варьироваться, но его всегда можно ограничить сверху как  $(d + 1) |\Omega_d^t|$ .

### Восстановление фона с помощью бесшовного монтажа

Наличие изменяющихся условий освещенности во входной видеопоследовательности или оставшиеся ошибки выравнивания между фрагментами фона, восстановленными с различных исходных кадров, могут приводить к образованию видимых швов в результате восстановления. Для снижения заметности швов в данной работе предлагается модификация метода бесшовного монтажа, известного как PoissonBlending [24]. Существующие обобщения этого метода для восстановления фона в видео [6] используют дополнительное слагаемое, накладывающее ограничение временной стабильности с учетом движения в восстановленной области. Отличительной особенностью предложенной модификации является использование адаптивных весов в слагаемом, отвечающим за поддержание временной



стабильности, что позволяет сохранять возможность равномерного изменения цвета всей восстанавливаемой области от кадра к кадру и, в то же время, усиливать ограничения временной стабильности в областях локально неконсистентного изменения цветов. А именно, предложенный алгоритм финального восстановления фона по вычисленному полю векторов  $V^t$  включает оптимизацию следующего функционала:

$$\begin{aligned}
 E(I^*) &= \quad (5) \\
 &= \sum_{p \in \Omega^t} \|\nabla I^*(p) - G^t(p)\|^2 + \sum_{p \in \Omega^t} w_p^{PB} \|I^*(p) - I^t(p)\|^2 + \\
 &+ \sum_{\substack{V^t(p) \neq V_{undef} \\ p \in \Omega^t}} (1 - w_p^{PB}) \|I^*(p) - I^{t_0}(p + O^{t \rightarrow t_0}(p))\|^2, \\
 w_p^{PB} &= \frac{1}{1 + \sigma_t \|\nabla I^{PB}(p) - G^t(p)\|^2}.
 \end{aligned}$$

Здесь, первое слагаемое штрафует отклонения поля градиентов результата от эталонного поля градиентов  $G^t$ , которое получается путем копирования градиентов из известных областей кадров в неизвестную область  $\Omega^t$ , согласно полю векторов  $V^t$ . Второе слагаемое – мягкая версия жесткого граничного условия, требующего совпадение результата и исходного кадра на границе неизвестной области  $\Omega^t$ . Третье слагаемое отвечает за поддержание временной стабильности относительно результата восстановления предыдущего обработанного кадра  $I^{t_0}$  с учетом восстановленного поля векторов движения  $O^{t \rightarrow t_0}$ . Компромисс между соблюдением граничных условий и поддержанием временной стабильности определяется весами  $w_p^{PB}$ , где  $I^{PB}$  – результат независимого применения метода PoissonBlending к текущему кадру (то есть, без учета третьего слагаемого и с  $w_p^{PB} \equiv 1$ ). Таким образом, граничные условия сменяются на ограничения временной стабильности в областях с высокой ошибкой воспроизведения эталонного поля градиентов  $G^t$ , где изменение цвета, требуемое для сокрытия швов, существенно локально непостоянно. В ходе экспериментов было установлено, что такая модификация кардинально повышает устойчивость метода к локальным выбросам на отдельных кадрах, обеспечи-

вая высокую временную стабильность. Однако, в отличие от кадрового применения алгоритма PoissonBlending, резко возникающие локальные швы могут сохранять заметность в результате восстановления фона.

Для снижения вычислительной сложности вместо минимизации  $E(I^*)$  напрямую проводится минимизация ее сепарабельной аппроксимации с использованием эвристики, предложенной в работе [25]. А именно, используя простое копирование согласно  $V^t$  для получения первого приближения восстановленного фона  $I_0$ , чередуется независимое построчное решение с игнорированием всех вертикальных зависимостей и независимое решение по столбцам:

$$\begin{aligned}
 I'_k &= \arg \min_{I^*} \left( \epsilon_k \sum_p \|I^*(p) - I_{k-1}(p)\|^2 + E_H(I^*) \right), \\
 I_k &= \arg \min_{I^*} \left( \epsilon_k \sum_p \|I^*(p) - I'_k(p)\|^2 + E_V(I^*) \right), \quad (6)
 \end{aligned}$$

$k = 1, \dots, K$

Здесь:  $E_H$  и  $E_V$  – модификации исходного функционала  $E$ , игнорирующие все вертикальные и горизонтальные зависимости между пикселями решения, соответственно. Вычислительная сложность минимизации таких модификаций значительно ниже, так как она состоит в решении СЛАУ с трехдиагональной матрицей, а не пятидиагональной, как в случае минимизации исходного функционала.  $\epsilon_k = \epsilon * a^{k-1}$  – экспоненциально возрастающие с номером итерации веса сохранения решения предыдущей итерации (в авторской реализации  $\epsilon = 0,0001$ ,  $a = 8$ , количество итераций  $K = 5$ ).

### Пространственное восстановление фона

Для восстановления фрагментов, которые нельзя восстановить копированием с какого-либо из кадров входной видеопоследовательности, то есть, фрагментов, которые не видны ни в одном из кадров входного видео, используется пространственное восстановление фона. После обработки всех кадров в прямом порядке, к последнему кадру применяется алгоритм, предложенный в работе [26], для обработки фрагментов фона, которые остались невосстановленными. Затем, при обработке кадров в обратном порядке, восстановленный на последнем кадре фон будет использоваться алгоритмом

Таблица 1. Значения показателя качества MSDSSIM на тестовой выборке (меньшие значения соответствуют лучшему результату). TeleaInpainting и PlanarStructure являются алгоритмами пространственного восстановления фона

Алгоритм	Среднее место	Тестовая последовательность						
		№ 1	№ 2	№ 3	№ 4	№ 5	№ 6	№ 7
Предложенный	1,9	0,267	0,222	0,091	<b>0,038</b>	<b>0,062</b>	<b>0,070</b>	<b>0,039</b>
BGR [21]	2,6	<b>0,221</b>	0,192	0,217	0,070	0,090	0,108	0,083
F_RigRemoval [29]	2,7	0,291	0,211	0,078	0,120	0,068	0,091	0,104
PFClean[30]	3,4	0,307	<b>0,187</b>	<b>0,077</b>	0,094	0,143	0,163	0,106
ComplexScenes [4]	5,3	0,307	0,252	0,116	0,162	0,195	0,355	0,237
TeleaInpainting [27]	5,7	0,333	0,623	0,614	0,206	0,141	0,133	0,367
PlanarStructure [28]	6,4	0,318	0,603	0,682	0,240	0,177	0,302	0,432

Таблица 2. Значения показателя качества  $C_{DSSIMdt}^{MS}$  на тестовой выборке (меньшие значения соответствуют лучшему результату). Данный показатель оценивает временную стабильность; методы пространственного восстановления (TeleaInpainting и PlanarStructure предсказуемо существенно проигрывают по этому параметру)

Алгоритм	Среднее место	Тестовая последовательность						
		№ 1	№ 2	№ 3	№ 4	№ 5	№ 6	№ 7
Предложенный	1,1	0,015	<b>0,007</b>	<b>0,004</b>	<b>0,005</b>	<b>0,007</b>	<b>0,005</b>	<b>0,004</b>
BGR [21]	2,7	<b>0,015</b>	0,007	0,024	0,009	0,009	0,010	0,007
PFClean [30]	3,0	0,017	0,007	0,004	0,011	0,015	0,013	0,009
F_RigRemoval [29]	3,6	0,020	0,015	0,006	0,014	0,008	0,009	0,011
ComplexScenes [4]	4,7	0,018	0,011	0,008	0,018	0,017	0,023	0,021
TeleaInpainting [27]	5,9	0,023	0,065	0,096	0,027	0,019	0,016	0,049
PlanarStructure [28]	7,0	0,071	0,105	0,128	0,060	0,039	0,081	0,091

Таблица 3. Время работы различных этапов предложенного алгоритма, описанных в соответствующих разделах статьи, и современного аналога на последовательности «satel» (90 кадров). Стоит отметить, что предложенный алгоритм работал в оригинальном разрешении 1920×1080, а время работы аналога указано авторами в статье [8] для уменьшенного разрешения 854×480

Алгоритм	Предложенный					Dynamic video [8]
	Формула (1)	Формула (4)	Формула (6)	Остальное	Целиком	
Время работы (сек.)	83,3	80,7	17,5	9,9	191,5	10800

как известная область, тем самым распространяя результат пространственного восстановления во времени. Аналогично, после обратного прохода алгоритма пространственное восстановление применяется к первому кадру видео. В результате, в ходе повторной обработки в прямом порядке восстановленным оказывается весь фон, находящийся в рамках кадра хотя бы на одном из крайних кадров входного видео. Для фрагментов фона, которые все еще остались не восстановленными после выполнения описанного алгоритма, применяется простой локальный метод пространственного восстановления [27].

### Экспериментальная оценка

Для экспериментальной оценки предложенного алгоритма использовался набор из 7 тестовых видеопоследовательностей с известными эталонными результатами и два показателя качества из предложенных в работе [9]. А именно, было проведено сравнение предложенного алгоритма с аналогами с использованием показателя качества MSDSSIM, который оценивает схожесть восстановленного фона с эталоном напрямую, тем самым показывая достоверность результатов восстановления. Для оценки временной стабильности был выбран показатель  $C_{DSSIMdt}^{MS}$ , как имеющий наиболее высокую корреляцию с результатами экспертной оценки среди всех показателей, проанализированных в [9].

Во всех проводимых экспериментах значения параметров предложенного алгоритма были фиксированы: параметр гладкости оптического потока  $\alpha = 2,5$ , параметр гладкости в шаге уточнении поля векторов  $\lambda = 200,0$ , параметр временной стабильности в бесшовном монтаже  $\sigma_t = 0,002$ . По результатам проведенного сравнения предложенный алгоритм превзошел

аналоги как по достоверности восстановленного фона (см. табл. 1), так и по временной стабильности (см. табл. 2). Дополнительно сравним результаты работы предложенного подхода и алгоритма, предложенного в работе [8], на нескольких примерах (см. рис. 2); также проводится сравнение по скорости работы (см. табл. 3). Тестирование предложенного алгоритма проводилось на системе с Intel i5-6600K 3.5 GHz.

Как видно из приведенных результатов количественной и качественной оценки предложенного алгоритма, он не уступает современным аналогам по качеству восстановления фона и существенно их превосходит по скорости работы.

### Заключение

В статье предложен новый алгоритм восстановления фона в видео, основанный на непараметрической модели движения фона и покадровом уточнении для предотвращения накопления ошибки, который не накладывает ограничений на допустимое движение камеры или объектов в сцене. Предложенный подход обладает существенно более низким временем работы, чем сопоставимые по качеству итогового результата аналоги, за счет вычисления оптического потока только в узлах разреженной сетки (1), использования приближенного итеративного метода решения СЛАУ с малым числом итераций при уточнении поля векторов (4) и сепарабельной аппроксимации в задаче бесшовного монтажа (6). В то же время, предложенный подход превзошел аналоги в проведенном объективном сравнении с алгоритмами с публично доступной реализацией и коммерческими инструментами для восстановления фона. Также, по качеству работы и широте класса допустимых входных видеопоследовательностей он сопоставим с современным подходом [8].



а) Сравнение на тестовой последовательности «camel»



б) Сравнение на тестовой последовательности «dance-twirl»



в) Сравнение на тестовой последовательности «rhino»

Рис. 2. Сравнение результата предложенного алгоритма и алгоритма, описанного в работе [8]. Авторы [8] дополнительно вручную модифицировали входные маски для удаления тени объектов

### Литература

1. J. Liu, G. Zhai, X. Yang, B. Yang, and L. Chen, «Spatial error concealment with an adaptive linear predictor», IEEE Transactions on Circuits and Systems for Video Technology, pp. 353–366, 2015.
2. Y. Matsushita, E. Ofek, W. Ge, X. Tang, and H. Y. Shum, «Full-frame video stabilization with motion

inpainting», IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1150-1163, 2006.

3. A. Sobral and A. Vacavant, «A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos», Computer Vision and Image Understanding, pp. 4-21, 2014.

4. A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez, «Video inpainting of complex scenes»,

SIAM Journal on Imaging Sciences, pp. 1993–2019, 2014.

5. Y. Shen, F. Lu, X. Cao, and H. Foroosh, «Video completion for perspective camera under constrained motion», IEEE International Conference on Pattern Recognition, pp. 63-66, 2006.

6. M. Granados, K. Kim, J. Tompkin, J. Kautz, and C. Theobalt, «Background inpainting for videos with dynamic objects and a free-moving camera», European Conference on Computer Vision (ECCV), pp. 682-695, 2012.

7. M. Roxas, T. Shiratori, and K. Ikeuchi, «Video completion via spatio-temporally consistent motion inpainting», IPSJ Transactions on Computer Vision and Applications, pp. 98-102, 2014.

8. J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, «Temporally coherent completion of dynamic video», ACM Transactions on Graphics (TOG), 2016.

9. Боков А.А., Ватолин Д.С. Методика объективной оценки качества восстановления фона в видео // Цифровая обработка сигналов. 2016. №3. С. 26-33.

10. Y. Wexler, E. Shechtman, and M. Irani, «Space-time video completion», IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2004.

11. T. Shiratori, Y. Matsushita, X. Tang, and S. B. Kang, «Video completion by motion field transfer», IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. 411–418, 2006.

12. C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, «Patchmatch: A randomized correspondence algorithm for structural image editing», ACM Transactions on Graphics (TOG), 2009.

13. K. Patwardhan and G. Sapiro, «Video inpainting under constrained camera motion», IEEE Transactions on Image Processing, pp. 545–553, 2007.

14. S.-C.-S. Cheung, J. Zhao, and M. V. Venkatesh, «Efficient object-based video inpainting», IEEE International Conference on Image Processing (ICIP), pp. 705-708, 2006.

15. M. Ebdelli, C. Guillemot, and O. Le Meur, «Exemplar-based video inpainting with motion-compensated neighbor embedding», IEEE International Conference on Image Processing (ICIP), pp. 1737-1740, 2012.

16. M. Ebdelli, O. Le Meur, and C. Guillemot, «Video inpainting with short-term windows: application to object removal and error concealment», IEEE Transactions on Image Processing, pp. 3034–3047, 2015.

17. M. Liu, S. Chen, J. Liu, and X. Tang, «Video completion via motion guided spatial-temporal global optimization», ACM International Conference on Multimedia, pp. 537-540, 2009.

18. N.-C. Tang, C.-T. Hsu, C.-W. Su, T.-K. Shih, and H.-Y.-M. Liao, «Video inpainting on digitized vintage films via maintaining spatiotemporal continuity», IEEE Transactions on Multimedia, pp. 602-614, 2011.

19. Z. Xu, Q. Zhang, Z. Cao, and C. Xiao, «Video Background Completion Using Motion-Guided Pixel Assignment Optimization», IEEE Transactions on Circuits and Systems for Video Technology, pp. 1393-1406, 2016.

20. M. Strobel, J. Diebold, and D. Cremers, «Flow and color inpainting for video completion», German Conference on Pattern Recognition (GCPR), pp. 293-304, 2014.

21. Зачесов А.А., Ерофеев М.В., Ватолин Д.С. Использование карт глубины при восстановлении фона в видеопоследовательностях // Новые информационные технологии в автоматизированных системах: материалы научно-практического семинара. 2015.

22. J. Herling and W. Broll, «High-quality real-time video inpainting with PixMix», IEEE Transactions on Visualization and Computer Graphics, pp. 866–879, 2014.

23. S. Liu, L. Yuan, P. Tan, and J. Sun, «Steadyflow: Spatially smooth optical flow for video stabilization», IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4209-4216, 2014.

24. P. Pérez, M. Gangnet, and A. Blake, «Poisson image editing», ACM Transactions on Graphics (TOG), pp. 313-318, 2003.

25. D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M.N. Do, «Fast global image smoothing based on weighted least squares», IEEE Transactions on Image Processing, pp. 5638-5653, 2014.

26. K. He and J. Sun, «Statistics of patch offsets for image completion», European Conference on Computer Vision (ECCV), pp. 16-29, 2012.

27. A. Telea, «An image inpainting technique based on the fast marching method», Journal of graphics tools, pp. 23–34, 2004.

28. J.-B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, «Image completion using planar structure guidance», ACM Transactions on Graphics (TOG), 2014.

29. The Foundry Nuke. <https://www.thefoundry.co.uk/products/nuke/>.

30. Pixel Farm PFClean. <http://www.thepixelfarm.co.uk/pfclean/>.

31. P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, «Deepflow: Large displacement optical flow with deep matching», IEEE International Conference on Computer Vision (ICCV), pp. 1385-1392, 2013.