

УДК 004.932

АЛГОРИТМИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ЭКСПЕРИМЕНТАЛЬНОЙ СИСТЕМЫ ВИДЕОНАБЛЮДЕНИЯ С ВОЗМОЖНОСТЬЮ УЛУЧШЕНИЯ ВИЗУАЛЬНОГО КАЧЕСТВА ВИДЕОПОТОКА

А.Г. Зотин, к.т.н.; А.И. Пахирка, к.т.н.; В.В. Буряченко

(Сибирский государственный аэрокосмический университет им. акад. М.Ф. Решетнева (СибГАУ), Красноярск, zotinkrs@gmail.com, pahirka@sibsau.ru, buryachenko@sibsau.ru)

Аннотация. В статье рассматривается реализация экспериментальной системы видеонаблюдения на основе IP-камер, предназначенной для изучения алгоритмов при организации видеонаблюдения на открытой местности. Наиболее значимыми факторами, оказывающими негативное влияние на восприятие видеoinформации, являются неравномерность освещения наблюдаемой сцены, погодные условия и дрожание камеры. Предлагаемая система предусматривает возможность улучшения качества видеопотока при помощи временных и пространственных методов. Для осуществления компенсации дрожания кадра предложен метод стабилизации видеопоследовательности, основанный на методе соответствия блоков с учетом пространственно-временной постобработки. В методе предусмотрен выбор предустановленных зон изображения, что позволяет повысить эффективность алгоритма при отсутствии движения камеры. Для оповещения оператора о происходящих в зоне наблюдения событиях предложен метод разделения движения на три уровня. Авторами предлагается нелинейное улучшение изображений с сохранением локального контраста в плохо и ярко освещенных областях. Применение нелинейной коррекции изображения позволяет повысить качество наблюдаемой сцены в вечернее время суток и при плохих погодных условиях. Представлены результаты экспериментов для детектора движения. Приведены данные об эффективности стабилизации и о повышении визуального качества изображений, полученных в сложных условиях освещения.

Ключевые слова: *видеонаблюдение, анализ видео, обнаружение движения, оценка движения, стабилизация видеопоследовательности, нелинейное улучшение качества, Retinex, HDR.*

Повышение эффективности современных систем видеонаблюдения позволяет снизить трудозатраты персонала. Однако в большинстве случаев применение таких систем ограничивается простой установкой видеокамер, при этом оператору необходимо постоянно следить за происходящим на нескольких камерах, данные с которых могут отображаться на множестве мониторов. Поэтому наряду с внедрением систем видеонаблюдения на автостоянки, заправочные станции, перекрестки со сложным движением требуется осуществлять аналитическую поддержку таких систем. Возникает потребность в использовании детекторов движения, в наличии удобного пользовательского интерфейса, позволяющего интуитивно следить за несколькими камерами, а также системы сигналов для оператора, привлекающей внимание к определенной ситуации на мониторе, в анализе движения объектов и их параметров. К тому же следует учесть, что при организации видеонаблюдения на открытой местности часто возникают проблемы, связанные с неблагоприятными погодными условиями.

Основным требованием к системам видеонаблюдения является ведение журнала событий в виде набора кадров или видеопоследовательностей, на которых запечатлены нарушители, с одновременным оповещением оператора. Функционирование блока, выполняющего эти действия, основано на работе детектора движения, однако в большинстве существующих систем используется детектор движения, интегрированный в камеры. Это приводит к тому, что необходимо очень тщательно подбирать параметры определения движения для снижения вероятности ложного срабатывания. При программной реализации детектора движения возможно использование адаптивных алгоритмов и алгоритмов, учитывающих предварительную обработку в виде стабилизации, цветовой и яркостной коррекции.

Предлагаемые алгоритмы предназначены для улучшения качества видеоматериала, полученного в сложных условиях освещения, при организации видеонаблюдения на АЗС, парковках, дворах и т.п. Трехступенчатый детектор движения позволяет выделить три вида движения в кадре: нет движения, слабое и сильное движение. В связи с проблемами детекторов движения, а именно большой чувствительностью к

погодным условиям, дрожанию камеры и незначительным помехам в кадре, предлагается алгоритм стабилизации видео, позволяющий значительно повысить устойчивость к таким эффектам. Применение временной фильтрации и метода MSR для повышения качества цветных изображений позволяет улучшить качество восприятия видеоданных в вечернее время суток, а также снизить влияние помех от самих камер.

Для разработки системы использовался модульный подход, поскольку применение модульной организации при реализации системы позволяет в дальнейшем с минимальными вложениями выполнять модернизацию или расширение системы в целом. Также модульная архитектура предоставляет более широкие возможности по отладке и изучению алгоритмов.

Основными модулями предлагаемой системы видеонаблюдения являются модули улучшения качества изображения, детектора движения и стабилизации видео. Кроме них, с ядром системы связаны модуль получения видеопотока, который получает данные с IP-камеры или из файла, а также модуль пользовательского интерфейса (рис. 1). Экранная форма исследовательского варианта экспериментальной системы видеонаблюдения представлена на рисунке 2.

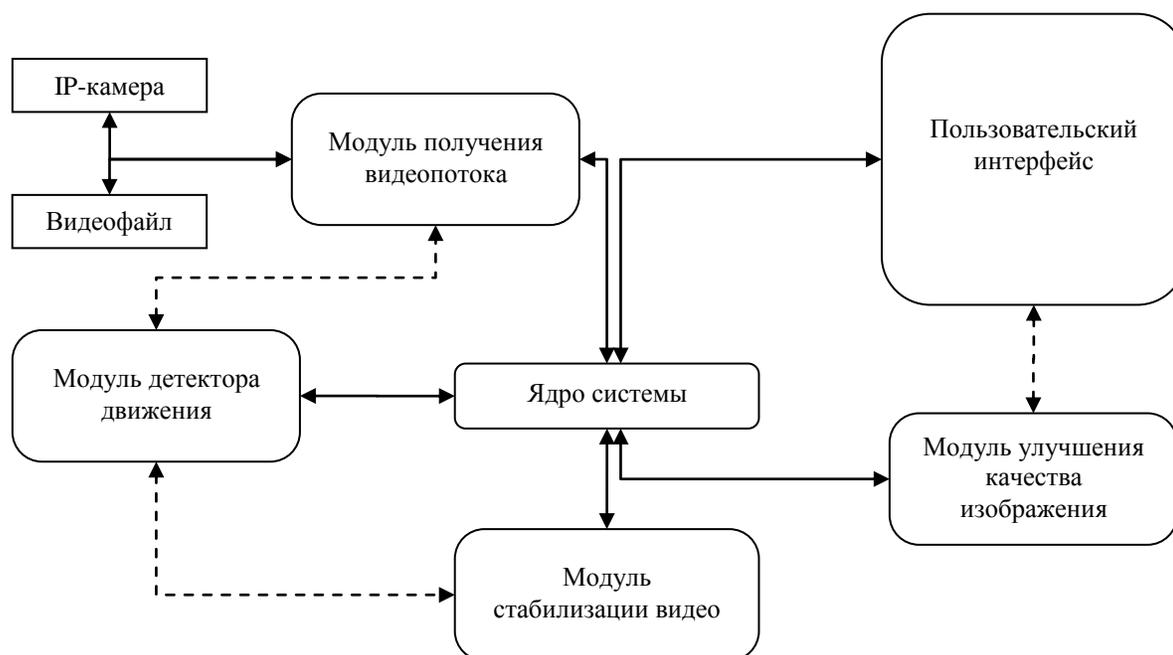


Рис. 1. Структура экспериментальной системы

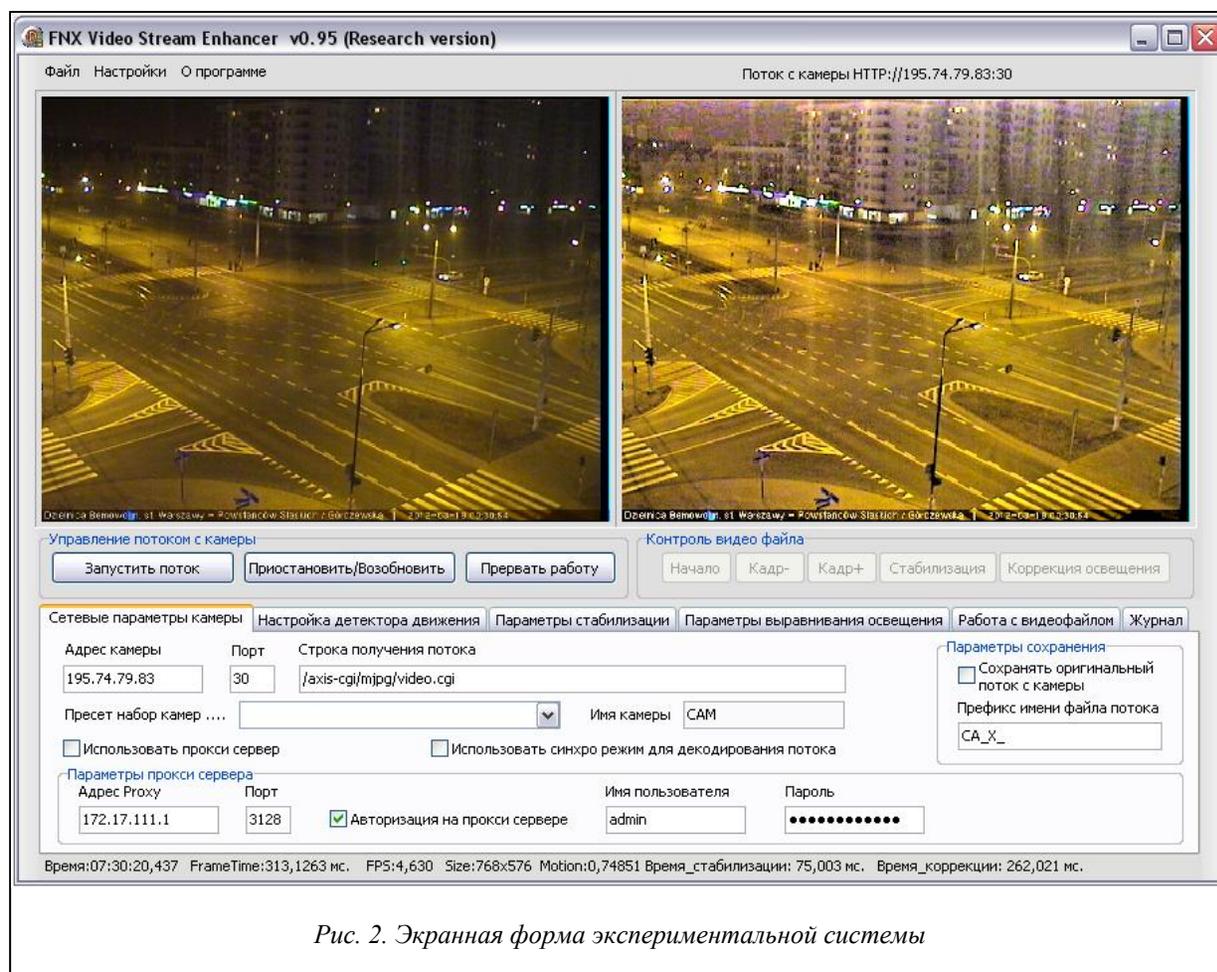


Рис. 2. Экранная форма экспериментальной системы

Оценка движения

Наиболее значимым в системах видеонаблюдения является модуль детектора движения, поскольку при обнаружении движения появляется необходимость в привлечении внимания оператора. На сегодняшний день существует большое количество методов обнаружения движения, однако многие из них достаточно сложны с вычислительной точки зрения. К самым быстродействующим методам определения движения относятся методы на основе вычитания фона и сравнения соседних кадров. В основе этих методов лежит сравнение текущего кадра с предыдущим или с некоторым опорным кадром. При этом для вычисления разницы между кадрами можно использовать три основных подхода к вычислению компонент: на основе компонент цветовой модели RGB, на основе яркости пикселей, на основе хроматических компонент цветовой модели YUV.

Для расчета величины движения был разработан метод, который предполагает использование факторов цветového усиления с использованием вычисления значения на основе цветовой модели RGB. Расчет абсолютной разности между значениями цветовых компонент R , G , B (Red, Green, Blue) пикселей анализируемых кадров I_{op} и I_{cur} проводится в нормированном диапазоне $[0..1]$ согласно формуле (1), с учетом расчета по цветовым компонентам, представленным в выражении (2). В качестве опорного кадра I_{op} можно использовать как предыдущий, так и фоновый кадр.

$$S_{dif}(s, x, y) = \begin{cases} 1, & \text{если } S_{ad}(s, x, y) \geq TS \times F_{boost}, \\ S_{ad}(s, x, y) \times K_{boost}, & \text{если } S_{ad}(s, x, y) \geq TS \text{ и } S_{ad}(s, x, y) < TS \times F_{boost}, \\ S_{ad}(s, x, y), & \text{если } S_{ad}(s, x, y) < TS, \end{cases} \quad (1)$$

где s – указатель компоненты цветовой модели (R, G, B); x и y – пространственные координаты; F_{boost} – величина, определяющая степень активации режима усиления; K_{boost} – коэффициент усиления; TS – пороговое значение, которое определяет, произошли ли изменения в цвете пикселя; $S_{ad}(s, x, y)$ – функция вычисления абсолютной разницы по указанной цветовой компоненте между заданными кадрами, которая вычисляется по формуле

$$S_{ad}(s, x, y) = |Sv_{top}(s, x, y) - Sv_{cur}(s, x, y)|, \quad (2)$$

где Sv_{top} , Sv_{cur} – значения компоненты цветовой модели (R, G, B) опорного и текущего кадров соответственно.

Для расчета величины изменения пикселя в нормированной форме цветовой модели RGB используется формула

$$P_{Diff}(x, y) = \frac{(S_{dif}(R, x, y) + S_{dif}(G, x, y) + S_{dif}(B, x, y))}{3}. \quad (3)$$

Зачастую шум в кадрах может быть засчитан как изменение в пикселе, и в итоге в большом количестве этот шум будет оказывать влияние на величину движения. Для уменьшения влияния шума на величину движения необходимо выполнить дополнительную коррекцию. Она осуществляется в зависимости от установленного порогового значения $[TS]$ и коэффициента подавления малых отклонений $[Lc]$ согласно формуле

$$Pl_{Diff}(x, y) = \begin{cases} 0, & \text{если } P_{Diff}(x, y) < TS \times Lc, \\ P_{Diff}(x, y), & \text{если } P_{Diff}(x, y) \geq TS \times Lc. \end{cases} \quad (4)$$

Параметр коэффициента подавления малых отклонений вычисляется на основе экспериментальных исследований и может быть скорректирован во время настройки параметров системы. Общая величина, характеризующая изменения пикселя $[PDiff(x, y)]$, рассчитывается по формуле

$$PDiff(x, y) = \begin{cases} Pl_{Diff}(x, y), & \text{если } Pl_{Diff}(x, y) < TS, \\ Pl_{Diff}(x, y) + 1, & \text{если } Pl_{Diff}(x, y) \geq TS \text{ и } Pl_{Diff}(x, y) < TS \times F_{boost}, \\ Pl_{Diff}(x, y) + 2, & \text{если } Pl_{Diff}(x, y) \geq TS \times F_{boost}. \end{cases} \quad (5)$$

Значение величины, характеризующей движение между двумя кадрами (изменения в кадре), вычисляется на основании общей величины, характеризующей изменение значений пикселей анализируемой области кадра, согласно формуле

$$F_{motion} = \frac{\sum_{x=0}^W \sum_{y=0}^H PDiff(x, y)}{H \times W} \times 100, \quad (6)$$

где W, H – размеры анализируемой области кадра по горизонтали и вертикали (ширина и высота) соответственно.

Для классификации движения используется значение величины, характеризующее движение между двумя кадрами, при этом следует отметить, что в зависимости от сцены диапазоны классификации движения могут варьироваться.

Алгоритм временного шумоподавления

Современные *CMOS*-матрицы *IP*-камер генерируют большое количество цифрового шума в зависимости от условий освещения и таких параметров, как чувствительность матрицы, качество матрицы и объектива, настройки параметров камеры. Уменьшить их влияние можно с помощью методов подавления шумов с минимальным искажением наблюдаемых объектов. Поскольку системы видеонаблюдения анализируют видеопоток, появляется возможность удаления шумов с использованием хронологии кадров (временная фильтрация) с последующей стабилизацией цветовых характеристик фонового изображения:

$$C_{new}(x, y) = \begin{cases} C_{cur}(x, y), & \text{если } |C_{cur}(x, y) - C_{old}(x, y)| \geq \gamma, \\ C_{old}(x, y) \times (1 - k) + C_{cur}(x, y) \times k, & \text{если } \delta < |C_{cur}(x, y) - C_{old}(x, y)| < \gamma, \\ C_{old}(x, y), & \text{если } |C_{cur}(x, y) - C_{old}(x, y)| \leq \delta, \end{cases} \quad (7)$$

где C_{old} – матрица значений компонентов цветовой модели для пикселей опорного кадра (по выбранной модели); C_{cur} – матрица значений компонентов цветовой модели для пикселей текущего кадра; C_{new} – матрица новых значений компонентов цветовой модели текущего кадра; δ – минимальный коэффициент выполнения коррекции (рекомендуется выбирать 1–2 % от динамического диапазона яркости); γ – верхний порог выполнения коррекции (рекомендуется выбирать 6–9 % от динамического диапазона яркости); k – коэффициент совмещения значений пикселей.

Стабилизация видеоданных

Модуль стабилизации видеопоследовательности нацелен на подавление нежелательной вибрации движения вследствие механического дрожания камеры и погодных условий, а также на синтез новых изображений последовательности с учетом стабилизации траектории движения камеры. При использовании стационарных *IP*-камер на открытом пространстве дрожание кадра может возникать при движении камеры либо из-за влияния ветра.

Для стабилизации видеопоследовательности необходимо провести оценку движения. В различных публикациях для оценки движения предложены методы, основанные на поиске особенных точек, методы оптического потока и методы соответствия блоков. Алгоритм *SIFT* выполняет поиск особенных точек, основываясь на вычислении гауссиана на изображении различного масштаба [1]. Он обнаруживает надежные точки и для каждой из них строит дескриптор, не зависящий от поворота, переноса и масштабирования. Частотные методы оценки движения в основном используются для глобальной оценки движения кадра. Наиболее известный из частотных методов – фазовый метод корреляции, который использует преобразование Фурье. Методы оптического потока гарантируют высокую точность для сцен с маленькими смещениями, но допускают сбои при больших смещениях. Алгоритм *SURF* основан на операции свертки и использует матрицу Гессе для вычисления меры распределения [2].

Большое число работ посвящено исследованию особенностей методов, основанных на методах соответствия блоков [3]: они используют различные фильтры для уточнения оценки движения и расчета локальных векторов движения. Эти алгоритмы обычно позволяют получить хорошие результаты, но могут давать ошибочные результаты для видеопоследовательностей, содержащих большие движущиеся объекты. Это происходит потому, что они не привязывают дескриптор к блоку и не отслеживают движение объектов на последовательности кадров. Для случая статичной камеры методы соответствия блоков подходят больше других, поскольку их применение позволяет отделить движение объектов в сцене от движения собственно камеры и наиболее быстро устранить нежелательное дрожание.

Схема блочной оценки движения содержит следующую последовательность действий. Изображение делится на неперекрывающиеся блоки пикселей B_{ij} , где i, j – координаты блока. Для каждого блока пикселей на предыдущем кадре ищется соответствующий блок на текущем кадре [2]. Задача оценки движения сводится к задаче поиска вектора движения $v_{i,j}$ для каждого блока $B_{i,j}$. При этом значение вектора $v_{i,j}$ определяется соотношением (8):

$$v_{i,j} = \arg \left[\min_{v_{i,j} \in \Omega} F_{(t,i,j,v_{i,j})} \right], \quad (8)$$

$$\Omega = \{x, y \mid x \in [-u_{\max}, u_{\max}], y \in [-v_{\max}, v_{\max}]\}, \quad (9)$$

где O – область поиска векторов движения, u_{max}, v_{max} – целые положительные числа; $F_{(i,j,v_i,j)}$ – мера близости блоков текущего и предыдущего кадров. Примером такой функции для оценки близости блоков является *SAD* (*Sum of Absolute Differences*), определяемая формулой

$$SAD(y, i, j, v) = \sum_{p \in B_{i,j}} I(p, t) - I(p, t) - I(p + v, t - 1) . \quad (10)$$

Результатом данного этапа являются рассчитанные для каждого блока изображения локальные векторы движения. Расчет глобального вектора движения производится на основе найденных локальных векторов по формуле

$$GVx = \frac{1}{MN} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} LVx(i, j), \quad GVy = \frac{1}{MN} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} LVy(i, j) . \quad (11)$$

После нахождения глобального вектора движения кадра выполняется компенсация движения. На данном этапе решаются две задачи: учет глобального движения кадра, а также восстановление границ, выпавших за рамки кадра [4]. Корректировка найденного дрожания кадра выполняется путем сдвига следующего кадра в направлении, обратном найденному вектору глобального движения. Таким образом, происходит смещение изображения на несколько пикселей, при этом необходимо учитывать изменение границы кадра. Чтобы избежать потери области видимости, часть информации для граничных пикселей берется из предыдущих кадров видеопоследовательности.

Последним этапом предлагаемого алгоритма стабилизации видео является постобработка кадра. Для нее используются автоматическая коррекция, а также обработка модифицированным для нескольких кадров фильтром *2d_cleaner*. В этом фильтре для каждого пикселя изображения рассчитывается значение с учетом окрестности по формуле

$$P_{x,y} = \begin{pmatrix} R = Ch(Ts, r) \\ G = Ch(Ts, r) \\ B = Ch(Ts, r) \end{pmatrix}, \quad (12)$$

где $P_{x,y}$ – пиксель, с окрестностью которого ведется работа; RGB – значения каналов RGB -спектра; Ts – значение порога, характеризующего возможность обработки; r – ранг окрестности; $Ch(Ts, r)$ – функция расчета значения канала в спектре:

$$Ch(Ts, r) = \frac{\sum_{i=-r}^r \sum_{j=-r}^r Sv(i, j)}{\sum_{i=-r}^r \sum_{j=-r}^r Cc(i, j)}, \quad (13)$$

где $Sv(i, j)$ – функция отсечения значения спектра по порогу; $Cc(i, j)$ – функция указания пригодности значения спектра по порогу, значение которой определяется по формуле (14).

$$Sv(i, j) = \begin{cases} sv_{i,j}, & \text{если } |sv_{i,j} - sv_{0,0}| \leq Ts, \\ 0, & \text{если } |sv_{i,j} - sv_{0,0}| > Ts, \end{cases} \quad Cc(i, j) = \begin{cases} 1, & \text{если } |sv_{i,j} - sv_{0,0}| \leq Ts, \\ 0, & \text{если } |sv_{i,j} - sv_{0,0}| > Ts. \end{cases} \quad (14)$$

где $sv_{i,j}$ – значение спектра рассматриваемого цветового канала; Ts – значение порога.

Реализация модифицированного фильтра *2d_cleaner* выполнена с учетом обработки по временной оси нескольких кадров видеопоследовательности, что позволяет уменьшить влияние помех после стабилизации. В ходе экспериментов было выяснено, что для видеопоследовательности, полученной со стационарной IP-

камеры, при выполнении такой фильтрации целесообразно использование двух или трех предыдущих кадров.

Нелинейное улучшение качества изображения

Поскольку в большинстве случаев при организации наружного видеонаблюдения встает задача улучшения визуального качества видеопотока для его лучшего восприятия оператором, в основу модуля улучшения качества изображений был положен модифицированный алгоритм *Multi-Scale Retinex (MSR)*, имитирующий визуальную систему человека [5]. Реализация алгоритма в виде отдельного модуля предоставляет возможность через ядро системы производить улучшение визуального качества изображений в полноформатном виде только для отображаемого на экране видеопотока.

MSR-алгоритм сжимает динамический диапазон изображения с сохранением (увеличением) локального контраста в плохо и ярко освещенных областях. Классический многомерный *MSR*-алгоритм [6] является взвешенной суммой одномерных *SSR*-алгоритмов (*Single-Scale Retinex*) для различных масштабов. Одномерная выходная функция i -го цветового канала $R_i(x, y, \sigma)$ вычисляется следующим образом:

$$R_i(x, y, \sigma) = \log\{I_i(x, y)\} - \log\{F(x, y, \sigma) * I_i(x, y)\}, \quad (15)$$

где $I_i(x, y)$ – входная функция i -го цветового канала по координатам x и y ; σ – масштабный коэффициент; $*$ – знак свертки функций; $F(x, y, \sigma)$ – гауссиан, определяемый как

$$F(x, y, \sigma) = K e^{-(x^2+y^2)/\sigma^2}. \quad (16)$$

При этом коэффициент K выбирается таким образом, чтобы выполнялось условие

$$\int \int_{\Omega_{x,y}} F(x, y, \sigma) dx dy = 1, \quad (17)$$

где $\Omega_{x,y}$ – множество пикселей, принадлежащих всему изображению.

Тогда многомерная выходная функция i -го цветового канала $R_M(x, y, w, \sigma)$ определяется как

$$R_M(x, y, w, \sigma) = \sum_{n=1}^N w_n R_i(x, y, \sigma_n), \quad (18)$$

где $w = (w_1, w_2, \dots, w_m)$, $m = 1, 2, \dots, M$ – весовой вектор одномерных выходных функций i -го цветового канала $R_i(x, y, \sigma)$; $\sigma = (\sigma_1, \sigma_2, \dots, \sigma_n)$, $n = 1, 2, \dots, N$ – вектор масштабов одномерных выходных функций. При этом

$$\sum_{n=1}^N w_n = 1.$$

Размерность вектора масштабов обычно выбирается не меньше 3. При проведении экспериментов они составили 13, 87, 180. Весовой вектор w , как правило, имеет элементы с равными значениями.

Однако классический алгоритм *MSR* приводит к искажению цвета изображения, так как значение каждой цветовой составляющей пикселя (например в *RGB*-пространстве) заменяется отношением ее исходного значения к среднему значению данной цветовой составляющей окружающих пикселей. Существует несколько решений данной проблемы. Так, некоторое улучшение результатов наблюдается при переходе в другие цветовые пространства с явным разделением яркостной и оттеночной составляющих (*HIS*-, *HSV*-, *HSL*-пространства). Однако лучший эффект достигается при использовании модели нормализованного разделения яркостной и оттеночной составляющих. Дополнительная обработка выполняется в соответствии с выражением

$$R'_M(x, y, w, \sigma, b) = R_M(x, y, w, \sigma) * I'_i(x, y, b), \quad (19)$$

где $I'_i(x, y, b)$ – нормализованная яркость, которая определяется по формуле

$$I'_i(x, y, c) = \log \left(1 + c \frac{I_i(x, y)}{\sum_{i=1}^3 I_i(x, y)} \right), \quad (20)$$

где c – коэффициент, выбираемый из середины диапазона значений $[0 \dots 255]$, $c=100 \div 125$.

Из-за характеристик логарифмической функции MSR-алгоритм делает детали изображения более различимыми в теневых областях, чем в засвеченных [5]. Чтобы сделать детали различимыми в засвеченных областях, можно применить логарифмическую функцию к инвертированному изображению. Строится модифицированная логарифмическая функция $L(I(x, y))$, зависящая от порогового значения Th , выбираемого пользователем:

$$L(I(x, y)) = \begin{cases} k_1 \cdot \log I(x, y), & I(x, y) < Th, \\ -k_2 \cdot \log(DR - I(x, y)) + \log DR, & I(x, y) \geq Th, \end{cases} \quad (21)$$

где

$$k_1 = \frac{Th \cdot \log DR}{\log Th}, \quad k_2 = \frac{\left(1 - \frac{Th}{DR}\right) \cdot \log DR}{\log DR - Th},$$

где DR – динамический диапазон изображения, в данном случае $DR = 255$ (для изображений с 8 битами на цветовой канал); k_1 и k_2 – весовые коэффициенты; Th – пороговое значение.

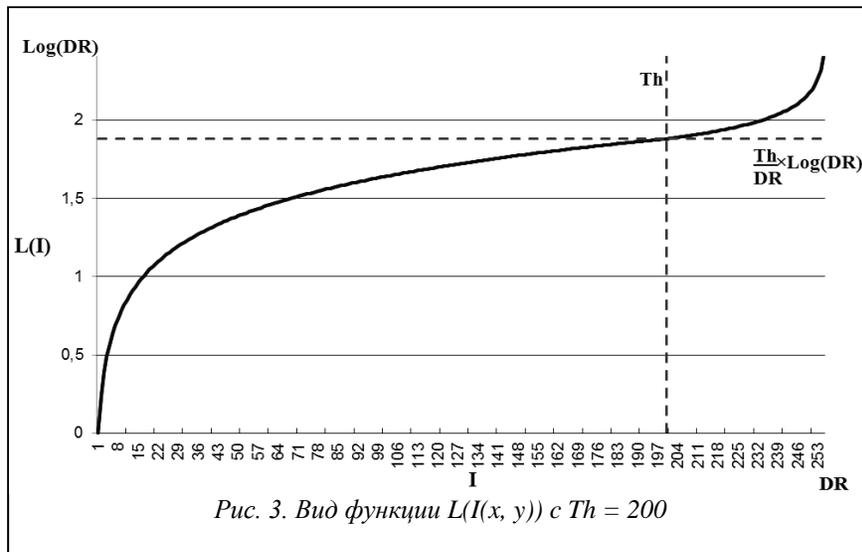


Рис. 3. Вид функции $L(I(x, y))$ с $Th = 200$

Пороговое значение Th было выбрано равным 200. Вид функции $L(I(x, y))$ приведен на рисунке 3.

Объединение ветвей дает классическому алгоритму MSR возможность повышать контрастность деталей в областях повышенной яркости. Использование такого вида логарифмической функции в определенной мере повышает контрастность деталей информативной области типичных распределений яркости (60–200), но в значительно меньшей мере, чем на затененных участках и участках с высокой яркостью.

Однако выходные данные MSR-алгоритма с модифицированной логарифмической функцией находятся вне динамического диапазона изображения [5]. Чтобы привести значения к динамическому диапазону изображения $[0 \dots 255]$, выполняется корректировка выходного изображения на основе корректирующего коэффициента и среднего значения яркостей пикселей обработанного изображения:

$$I_{enh}(x, y) = \frac{I_m(x, y) - I_{avg}(x, y)}{w_{cor}} + C, \quad (22)$$

где $I_m(x, y)$ – яркость пикселя после обработки MSR-алгоритмом с модифицированной логарифмической функцией; $I_{avg}(x, y)$ – среднее значение яркости пикселей обработанного изображения; w_{cor} –

корректирующий коэффициент (выбирается экспериментальным путем ($w_{\text{cor}} = 0,007$), за константу C берется половина динамического диапазона изображения ($C = 127,5$)).

На рисунке 4 показан пример работы *MSR*-алгоритма с модифицированной логарифмической функцией. Сравнительный пример работы предлагаемого алгоритма с классическим методом *MSR*, реализованным в редакторе *GIMP*, показан на рисунке 5, красной линией обведена область с высокой яркостью, где, в отличие



от классического метода *MSR*, также повышается контрастность деталей изображения.



Рис. 5. Пример нелинейного улучшения изображений: а) входное изображение, б) изображение с применением модифицированного метода MSR, в) изображение с применением классического метода MSR в редакторе GIMP

Другие алгоритмы улучшения качества изображений

Существует множество алгоритмов улучшения визуального качества изображений, в числе самых распространенных – алгоритмы гамма-коррекции (*gamma correction*) и выравнивания гистограммы

изображения (*histogram equalization*). Гамма-коррекция представляет собой изменение яркости изображения с использованием степенной функции:

$$I_{out}(x, y) = I_{inp}(x, y)^{\gamma}, \quad (23)$$

где $I(x, y)$ – яркость пикселя; γ – коэффициент коррекции, обычно выбирается от 1,8 до 2,4 (в нашем случае использовался коэффициент 2,2, рис. 6б).

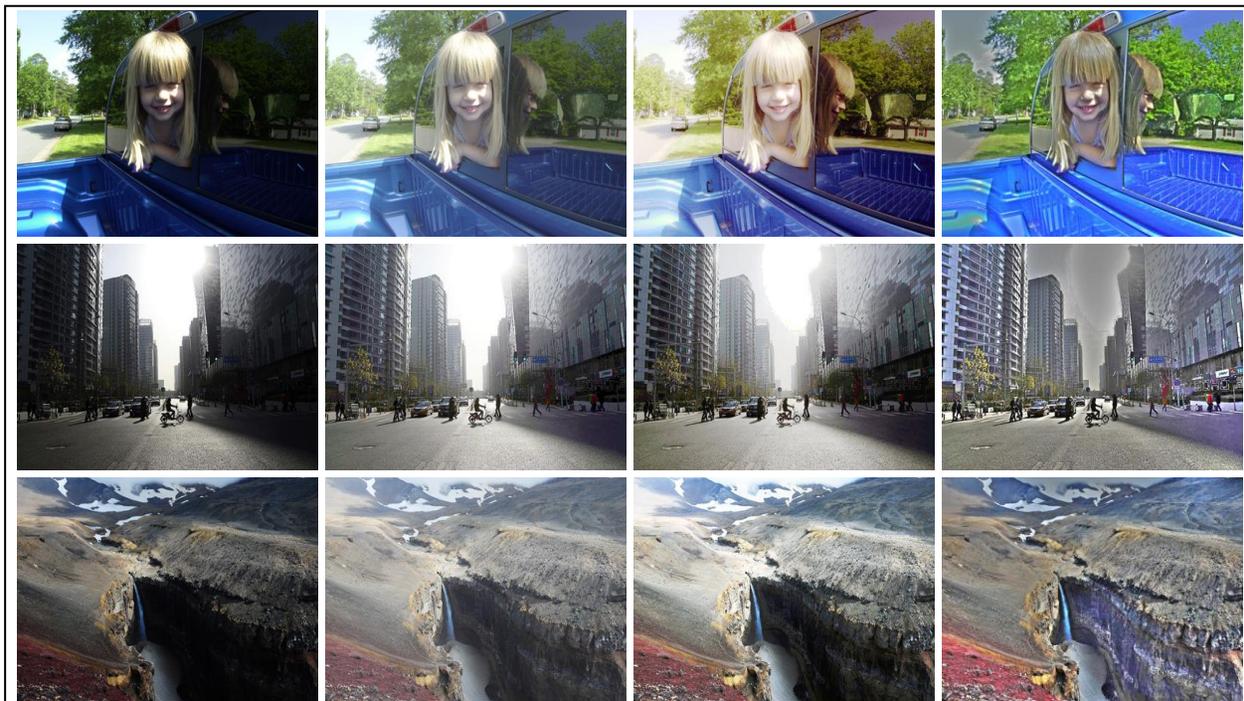


Рис. 6. Пример работы алгоритмов улучшения изображений: а) входное изображение, б) гамма-коррекция изображения, в) выравнивание гистограммы изображения, г) изображение с применением MSR-алгоритма

Алгоритм выравнивания гистограммы увеличивает общий контраст изображения в соответствии со следующим выражением:

$$w_y(y) = \begin{cases} \frac{1}{y_{\max} - y_{\min}}, & y_{\min} \leq y \leq y_{\max}, \\ 0, & \text{при других } y, \end{cases} \quad y = (y_{\max} - y_{\min})F_x(x) + y_{\min}, \quad (24)$$

где y_{\max} , y_{\min} – максимальное и минимальное значения яркости изображения; $F(x)$ – функция распределения яркостей входного изображения.

Для получения функции распределения яркостей измеряется гистограмма исходного изображения. Для цифрового изображения, шкала яркостей которого принадлежит диапазону $[0..255]$, гистограмма представляет собой таблицу из 256 чисел. Каждое из них показывает количество точек в изображении (кадре), имеющих данную яркость. Пример работы алгоритма выравнивания гистограммы изображения показан на рисунке 6в.

Экспериментальные результаты

Для оценки эффективности и целесообразности использования предложенных методов было разработано экспериментальное программное обеспечение. Проверка методов обнаружения движения и сбор статистики осуществлялись с использованием видеоданных с девяти IP-камер, расположенных как на оживленных улицах, так и в местах с минимальным движением. Исследование проводилось на 57 последовательностях по 1500 кадров в каждой.

Для оценки алгоритма временного шумоподавления решено использовать метрику Δ , наглядно показывающую отличия между кадрами. Вычисление метрики выполняется согласно формуле

$$\Delta(x, y) = \begin{cases} R |C_{cur}(x, y) - C_{prev}(x, y)|, & \text{если } C_{cur}(x, y) - C_{prev}(x, y) < 0, \\ G |C_{cur}(x, y) - C_{prev}(x, y)|, & \text{если } C_{cur}(x, y) - C_{prev}(x, y) > 0, \\ 0, & \text{если } C_{cur}(x, y) - C_{prev}(x, y) = 0, \end{cases} \quad (25)$$

где C_{cur} – матрица значений компонентов цветовой модели для пикселей текущего кадра (по выбранной модели); C_{prev} – матрица значений компонентов цветовой модели предыдущего кадра; $R()$, $G()$ – функции, устанавливающие значение в цветовом канале *Red* и *Green* модели *RGB*.

Результаты работы алгоритма временного шумоподавления с яркостной коррекцией для фрагмента видеопотока, полученного с камеры на перекрестке в городе Берн (Швейцария), показаны на рисунке 7. В ходе экспериментов было выяснено, что для динамического диапазона изображения [0...255] наиболее подходящими значениями коэффициентов будут $\delta=5$, $\gamma=16$, $k=0,4$.

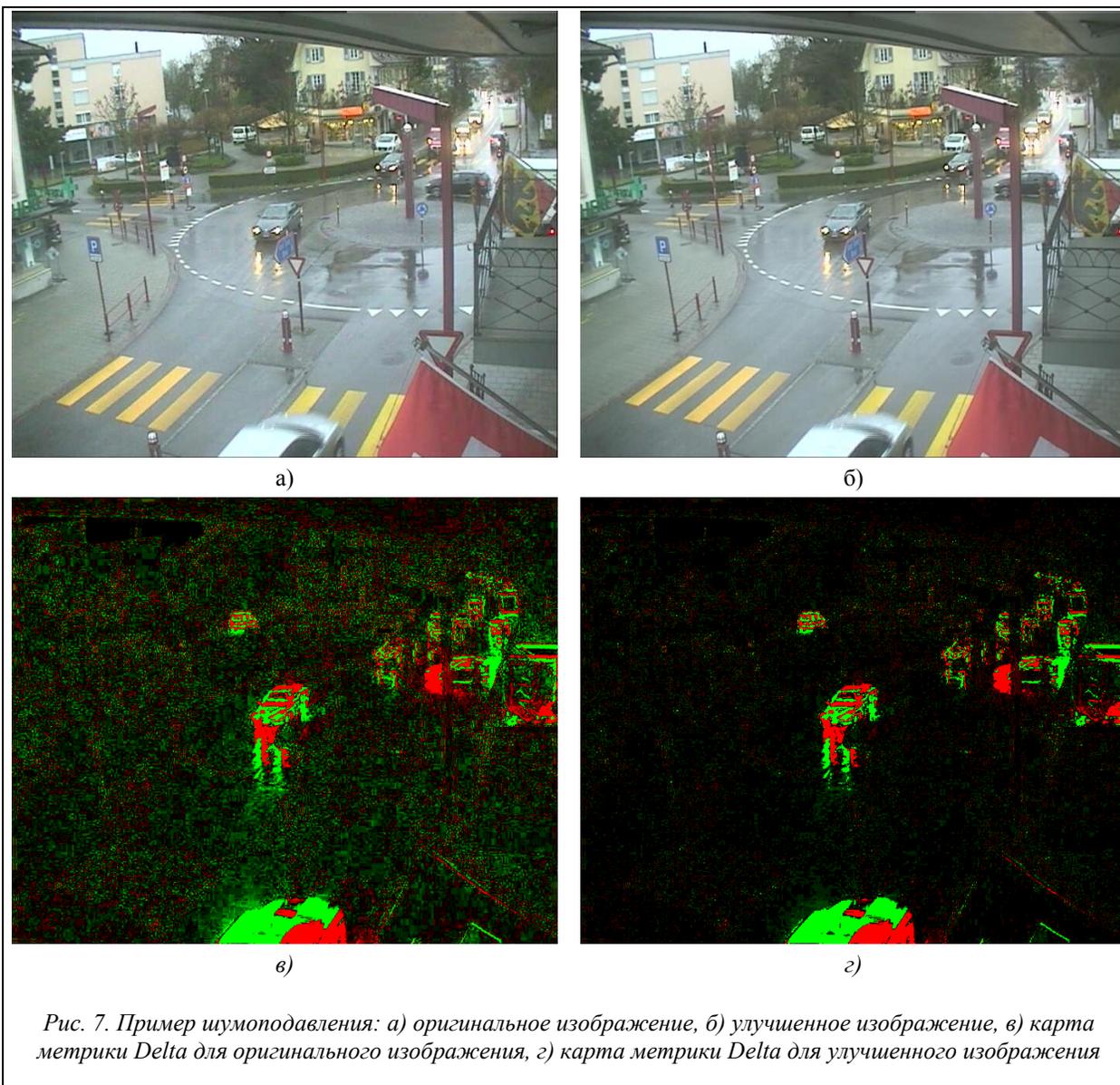


Рис. 7. Пример шумоподавления: а) оригинальное изображение, б) улучшенное изображение, в) карта метрики Delta для оригинального изображения, г) карта метрики Delta для улучшенного изображения

В разработанном методе сравнение движения проводилось при одинаковом способе вычисления величины движения, отличие заключалось в разных способах формирования значения абсолютной разницы, то есть в использовании трех основных подходов. На рисунке 8 представлен график, характеризующий величину движения для разных реализаций метода обнаружения движения на отрезке видеопоследовательности с двумя интервалами движения. Фрагмент последовательности получен с IP-камеры, находящейся на берегу залива Nordfjorden (Норвегия).



Наибольшую эффективность предложенный метод показал в цветовой модели *RGB* с использованием фактора цветowego усиления, равного 4. Для выяснения возможности ускорения работы алгоритма вычисления величины движения был проведен ряд экспериментов. Для оценки быстродействия и точности работы детектора движения использовались кадры различного масштаба, так как уменьшение анализируемого изображения дает существенное увеличение скорости работы. В таблице 1 представлены данные о точности определения движения и уровне ложного определения движения. На рисунке 9 представлены результаты анализа быстродействия работы детектора движения в зависимости от используемого масштаба кадров. Данные результаты были получены в следующих условиях: размер кадра 640×480 на компьютере со следующими параметрами: процессор Intel Core 2 duo 3 ГГц, ОЗУ 4Gb, видеокарта

Таблица 1

Качество детектирования движения в зависимости от масштаба кадров

Детектирование	Масштаб кадров							
	100 %	70 %	60 %	50 %	40 %	30 %	20 %	10 %
Точность определения движения (%)	94,31	95	94,5	93,77	95,01	94,27	88,4	78
Ложное определение движения (%)	2,78	3,19	3,6	3,28	5,6	4,1	7,8	23,8

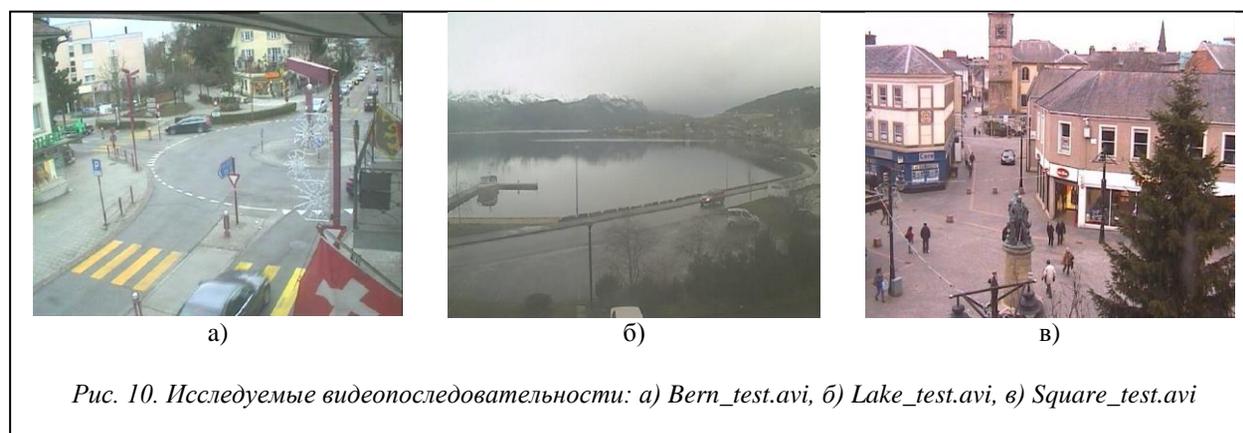
Nvidia Geforce GTS 250, операционная система Windows 7.

Тестирование разработанного алгоритма стабилизации проводилось на 15 видеопоследовательностях, полученных с доступных в открытом доступе *IP*-камер. Результаты оценки качества стабилизации для трех видеопоследовательностей, полученных с различными условиями съемки (рис. 10):

- различные объекты интереса на переднем и заднем планах изображения;
- изменение положения камеры, объекты интереса на заднем плане кадра, низкая освещенность;
- темное время суток, высокая зашумленность кадра.



Объективная оценка качества стабилизации видеопоследовательности выполняется с использованием метрики: пиковое отношение сигнала к шуму (*PSNR*) как мера ошибки. *PSNR* между кадрами k и $k-1$



определяется по формуле

$$PSNR(k) = 10 \log_{10} \left(\frac{I_{\max}^2}{MSE(k)} \right), \quad (26)$$

где $MSE(k)$ – среднеквадратичная межкадровая ошибка; I_{\max} – максимальное значение интенсивности пикселя; M, N – размеры кадра.

Значение *PSNR* показывает, насколько одно изображение похоже на другое, что полезно для оценки качества стабилизации простым сравнением соседних кадров. Качество межкадрового преобразования *Inteframe Transformation Fidelity (ITF)* используется для объективной оценки эффективности алгоритма стабилизации: стабилизированная видеопоследовательность должна иметь более высокое значение *ITF*, чем оригинальная:

$$ITF = \frac{1}{N_{Frame}} \sum_{k=0}^{N_{Frame}} PSNR(k), \quad (27)$$

где N_{Frame} – количество кадров видеопоследовательности.

В таблице 2 приведены результаты оценки качества стабилизации ITF , рассчитанной с применением метрики $PSNR$ для представленных на рисунке 10 видеопоследовательностей. Для сравнения использовались *VirtualDub MSU Deshaker filter* и фильтр для стабилизации *Warp Stabilizer*, встроенный в *Adobe After Effects*

Таблица 2

Сравнение эффективности систем стабилизации

Видеопоследовательность	Значение ITF, dB		
	Deshaker	WarpStabilizer	Предлагаемая реализация
Bern_test.avi	32,31	31,48	36,1
Lake_test.avi	38,53	38,7	39,14
Square_test.avi	36,65	33,41	36,15

CS 5.5.

В зависимости от типов движения, присутствующих на видеопоследовательности, достигаются различные результаты стабилизации. Предлагаемый алгоритм дает наилучшие результаты при наличии возможности указать области, где отсутствует резкое движение объектов, что позволяет исключить ошибки в нахождении траектории движения камеры.

В ходе оценки качества стабилизации с применением пространственно-временного фильтра *2D Cleaner* выяснено, что при стабилизации видеоматериала модифицированная реализация *2D Cleaner* значительно улучшает качество, устраняя следы мелкого дрожания видеопоследовательности, от которого не удалось избавиться при стабилизации (рис. 11). $PSNR$ -значение стабилизированной последовательности заметно выше, что показывает меньшую разницу между кадрами (более низкое дрожание кадра). Также при применении временного фильтра $PSNR$ -значение видеопоследовательности увеличивается в пределах 30 %.

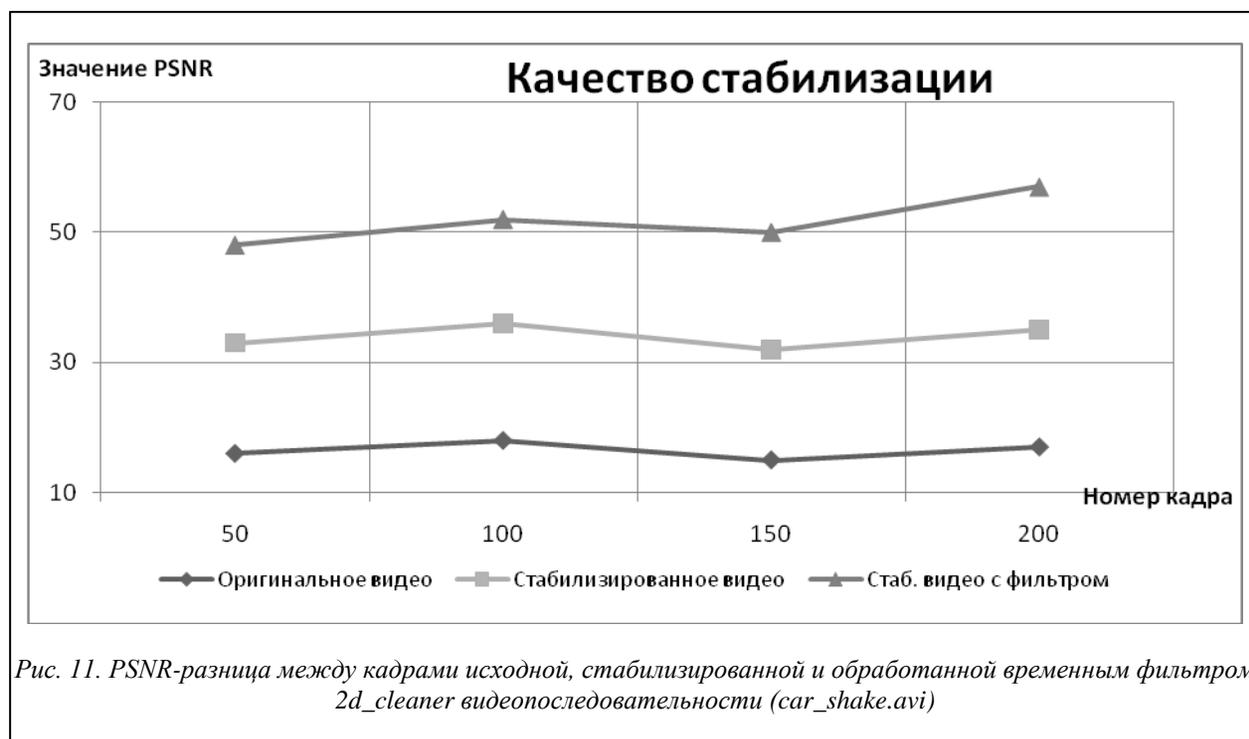
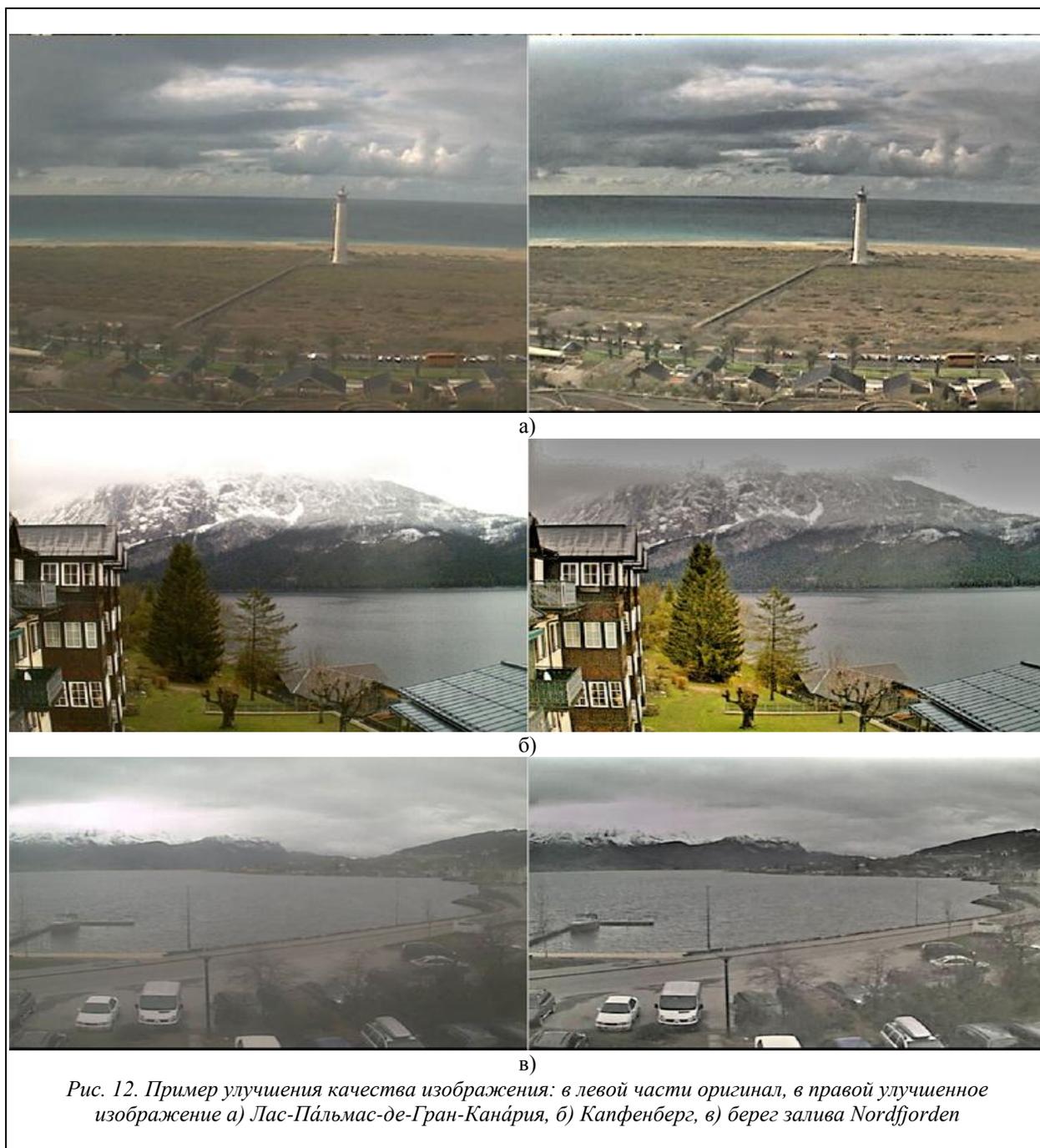


Рис. 11. PSNR-разница между кадрами исходной, стабилизированной и обработанной временным фильтром *2d_cleaner* видеопоследовательности (*car_shake.avi*)

Проверка модуля улучшения визуального качества выполнялась на последовательностях, содержащих искажения, вызванные различными природными явлениями и условиями освещения. Примеры работы модуля показаны на рисунке 12. Представлены фрагменты с камер, расположенных в Лас-Пальмас-де-Гран-Канария (Испания) (съемка происходит в пасмурную погоду), в Капфенберге (Австрия) (съемка в нормальную погоду, наблюдается закат в области гор), на берегу залива Nordfjorden (Норвегия) (съемка в туманную погоду).



Предлагаемая система видеонаблюдения с возможностью улучшения визуального качества позволит повысить эффективность работы службы охраны благодаря возможностям улучшения визуального качества отображаемых данных и стабилизации видеопотока. Предложенные детектор движения и временная фильтрация позволяют более качественно оценивать ситуацию в наблюдаемой зоне. При изменении масштаба от 100 % до 30 % фактор движения, характеризующий точность определения движения, находится на близком уровне, а при масштабе менее 15 % значительно ухудшается. Использование стабилизации видеопотока позволяет повысить качество нахождения движения в среднем на 2–7 % в зависимости от

степени дрожания камеры, поскольку оно оказывает значительное влияние на работу алгоритма, а также улучшить возможности визуального наблюдения для оператора.

Литература

1. Lowe D. Intern. Journ. of Computer Vision, 2004, 60(2), pp. 91–110.
2. Bay H., Ess A., Tuytelaars T., Van Gool L., Computer Vision and Image Understanding, 2008, Vol. 110, pp. 346–359.
3. Rawat P., Singhai J., SIPIJ, 2011, Vol. 2, no. 2, pp. 159–168.
4. Ko-Cheung H., Wan-Chi S., IEEE Transactions on Image Processing, 2007, Vol. 16 (5), pp. 1232–1245.
5. Фаворская М.Н., Зотин А.Г., Пахирка А.И. Метод улучшения цветных изображений на основе выравнивания спектральных диапазонов и коррекции контрастности // Механика, управление и информатика. М.: ИКИ РАН, 2012, № 8, с. 99–103.
6. Choi D.H., Jang I.H., Kim M.H., Kim N.C., Proc. IEEE Int. Symp. Circuits and Syst., New Orleans, USA, 2007, pp. 3948–3951.