

К. Ю. Дергачёв, Л. А. Краснов, А. А. Челядин, А. В. Плахотний

Национальный аэрокосмический университет имени Н. Е. Жуковского «ХАИ», Харьков, Украина

РАЗРАБОТКА МЕТОДОВ И СРЕДСТВ ЦВЕТОВОЙ КОРРЕКЦИИ WEB-КАМЕР В СИСТЕМАХ БИНОКУЛЯРНОГО ЗРЕНИЯ

Предмет изучения. В статье предложены новые методы и средства цветовой коррекции web-камер в системах стереозрения для повышения качества их работы. **Целью** является сопоставительный анализ показателей качества известных методов коррекции цвета, разработка новой методики и рабочих алгоритмов совместной процедуры цветовой коррекции и ректификации фреймов видеопотока левой и правой камер. **Задачи:** Ставилась задача выполнить теоретический анализ показателей качества известных алгоритмов коррекции цвета, разработать новые рабочие алгоритмы, программные коды этих алгоритмов написать на языке Python с использованием необходимых функций OpenCV. Провести экспериментальные исследования этих алгоритмов. Оценить эффективность работы стереосистемы в лабораторных условиях, а достоверность полученных результатов проверить методами статистического анализа. **Используемые методы:** Сопоставительный анализ известных методов и алгоритмов путем статистического моделирования, синтез новых алгоритмов и оценка эффективности их работы путем проведения лабораторных натурных испытаний. **Полученные результаты:** проведен сопоставительный анализ эффективности работы известных методов цветовой коррекции камер стереосистемы, предложены новые более эффективные алгоритмы для решения этой задачи. **Выводы.** Научная новизна полученных результатов: предложены новые методы коррекции цветового баланса web-камер, используемых в стереоскопических системах технического зрения, отличающиеся высокой точностью цветовой коррекции и работающие в реальном масштабе времени с применением функций библиотеки OpenCV в программной среде Python.

Ключевые слова: стереоскопические системы технического зрения; цветровая коррекция изображений видеопотока web-камер стереосистемы; совместная ректификация и цветовой баланс левой и правой камер.

Введение

Массовое внедрение инновационного оборудования (роботов, беспилотных летательных аппаратов и пр.) в различные области человеческой деятельности в настоящее время заметно повысило спрос на недорогие, но качественные системы технического зрения. Так, например, существенным ограничением при создании малобюджетных систем стереосъемки является стоимость откалиброванных камер даже невысокого разрешения. Оптическая калибровка требует большой точности изготовления элементов камер. Это заметно удорожает их производство и создаёт проблемы с регулировкой размеров стереобазы. Наличие же широкодоступных и дешевых некалиброванных web-камер с возможностью их подключения к компьютеру через USB-порты позволяет синтезировать недорогие и достаточно точные системы стереосъемки с легко изменяемыми параметрами.

Однако следует отметить, что даже незначительные рассогласования параметров цифровых стереоскопических измерителей приводят к существенному ухудшению качества их работы, вплоть до исчезновения стереоэффекта. Анализ этих факторов [1, 2] наглядно показывает, что любые несоответствия пространственно-временных характеристик и параметров цветопередачи приводят к деформации измерительного пространства и значительному уменьшению точности измерений цифровых бинокулярных систем технического зрения. Поэтому перед началом эксплуатации для формирования высококачественных изображений стереопары, необходимо выполнить ряд следующих мероприятий:

- сделать внутреннюю калибровку используемых камер для определения их параметров – фокусного расстояния, оптических искажений и пр.;

- произвести программную синхронизацию работы камер с большой точностью для получения синхронизированных стереоизображений с высокой частотой повторения;

- осуществить стереокалибровку камер в общей системе координат;

- провести ректификацию изображений для установления соответствий между эпиполярными линиями на стереопаре, вдоль которых будет выполняться поиск для работы алгоритмов стереосопоставления;

- выполнить цветовую коррекцию стереоканалов.

Обычно, специалисты в области технического зрения считают, что различная яркость и баланс белого стереоканалов мало влияют на точность при решении задач стереосопоставления, но они пагубны при стереовизуализации. По мнению авторов, различная интенсивность компонент цвета может не только существенно исказить зрительное восприятие (вплоть до потери эффекта трёхмерности), но и фатально повлиять на качество определения параметров пространственного положения фрагментов наблюдаемой сцены. Перечисленные выше работы сравнительно легко выполняются достаточно традиционными методами. Так, например, внутреннюю калибровку используемых web-камер можно провести с помощью Matlab-приложения Camera Calibration Toolbox for Matlab [3] или привлечь соответствующие ресурсы библиотеки OpenCV [4]. Однако существует и ряд нерешенных проблем. Так наибольшая сложность прецизионной настройки системы стереозрения связана с отсутствием единого подхода к выбору методов и средств для цветовой коррекции камер и процедуры их ректификации, что делает эту задачу неизменно актуальной.

Цель работы. Для повышения качества работы стереосистем ставились следующие задачи: провести сопоставительный анализ показателей качества известных методов коррекции цвета, предложить новую методику и рабочие алгоритмы совместной процедуры цветовой коррекции и ректификации фреймов видеопотока левой и правой камер. Программные коды этих алгоритмов было необходимо написать на языке Python с использованием необходимых функций OpenCV. Следовало представить подробные результаты экспериментальных исследований алгоритмов цветовой коррекции видеоданных стереопары. Оценка эффективности работы стереосистемы была проведена в лабораторных условиях, а достоверность полученных данных проверялась путем статистического анализа полученных данных.

Анализ методов цветовой коррекции камер стереопары

Цветовую коррекцию камер стереопары так же, как и их пространственную калибровку, по мнению авторов, целесообразно проводить в два этапа. Сначала необходимо точно установить баланс белого и устранить различия интенсивности компонент цвета изображений видеопоследовательностей каждой из камер отдельно, а затем провести совместную корректировку этих параметров с учетом результатов взаимной ректификации камер. Далее такой подход рассмотрим более подробно.

Оценим основные свойства известных методов выполнения цветовой коррекции. Их актуальность обычно обусловлена локальными и глобальными несоответствиями цветов между камерами в составе стереопары [5]. Кроме этого, существенным фактором является наличие или отсутствие бликов на объектах при наблюдении за ними с разных ракурсов. Основные причины такого рода несоответствий обусловлены различиями фоточувствительных матриц и светофильтров камер стереопары, и их неодинаковым положением относительно источников освещения сцены. Глобальные методы цветовой коррекции можно разделить на три основных класса: histogram matching; global color transfer; clustering method. При классификации локальных способов цветовой коррекции изображений уместно выделить два основных метода – block-based method и local color correction for underwater stereo images. Рекомендуется при выборе любого из перечисленных методов цветовой коррекции дополнять такую обработку алгоритмами обнаружения и компенсации бликов. При использовании методов histogram matching работа алгоритмов цветовой коррекции в основном сводится к расслоению снимков в RGB-пространстве на отдельные компоненты R, G и B, с последующим построением гистограмм распределения пикселей по яркости для левого и правого ракурсов стереоизображений. Затем формируется функция преобразования для одного из ракурсов для получения минимального различия этих гистограмм для каждой цветовой компоненты.

Методы global color transfer основаны на синтезе такой линейной функции преобразования цветового RGB-пространства, чтобы целевое изображе-

ние имело такие же цветовые тона, что и у исходного изображения. В этом случае проводится вычисление средних значений пикселей по каждому из каналов R , G , B для исходного и целевого изображений, а затем с помощью матриц сдвига, поворота и масштабирования проводится необходимое совместное преобразование по трем цветовым каналам.

Применение clustering method коррекции основано на выборе ключевых кадров и проведении вероятностной кластеризации пикселей с последующим вычислением параметров цветовой коррекции и их интерполяции в неключевых кадрах. Используются методы статистической обработки данных.

Аналогичные принципы применяются при построении локальных методов цветовой коррекции и алгоритмов устранения бликов [5, 6].

К сожалению, описанные методы цветовой коррекции изображений, обладающие рядом несомненных достоинств, отличаются большим объемом вычислений и малоприспособлены для работы в реальном масштабе времени. Поэтому целесообразно исследовать возможности оптимизации структуры таких алгоритмов для создания программного обеспечения, позволяющего быстро и эффективно проводить цветовую коррекцию камер стереопары.

Средства построения алгоритмов для коррекции цвета отдельной камеры

Как отмечалось ранее, корректировку баланса белого (цветового баланса) целесообразно сначала провести отдельно для каждой камеры стереопары, а затем скорректировать полученные результаты уже для пары камер. Известны три основных метода балансировки белого для цифровых камер: автоматическое выравнивание баланса; установки для набора стандартных условий освещения; произвольные установки по опорному объекту.

Рассмотрим подробнее реализацию основных алгоритмов для каждого из перечисленных методов.

Алгоритм «серого мира». Наиболее популярным методом автоматической цветовой коррекции изображений является метод, получивший название «Серый мир», поскольку предполагается, что сумма всех цветов на изображении естественной сцены дает серый цвет. При использовании такого метода в пространстве проводится разложение исходного изображения I_{src} размерностью $M \times N$ на три компоненты $(R_{src}, G_{src}, B_{src})$. Здесь и далее I_{src} – исходное изображение (image source), а I_{dst} – выходное изображение (image destination). Затем рассчитываются средние значения яркости пикселей для каждой из этих компонент:

$$\bar{R} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N R_{src}(x_i, y_j), \quad \bar{G} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N G_{src}(x_i, y_j),$$

$$\bar{B} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N B_{src}(x_i, y_j),$$

где x_i и y_j – текущие номера строк и столбцов исходного изображения I_{src} . Затем определяется

обобщенный коэффициент средней яркости пикселей MB (Medium Brightness) как $MB = (\bar{R} + \bar{G} + \bar{B})/3$ и производится масштабирование пикселей исходного изображения сцены I_{src} с соответствующими коэффициентами для получения $(R_{dst}, G_{dst}, B_{dst})$ – компонент выходного изображения I_{dst} :

$$R_{dst} = R_{src} \cdot \frac{MB}{R}, \quad G_{dst} = G_{src} \cdot \frac{MB}{G}, \quad B_{dst} = B_{src} \cdot \frac{MB}{B}.$$

После корректировки эти компоненты соединяются в цветное изображение I_{dst} , которое легко сравнить с исходным изображением I_{src} визуально, а для дополнительной наглядности используются изображения разницы выходной и исходной яркостей всех пикселей для соответствующих цветовых компонент, полученных в результате преобразования:

$$\Delta R = R_{dst} - R_{src}, \quad \Delta G = G_{dst} - G_{src}, \quad \Delta B = B_{dst} - B_{src}.$$

Для описания степени соответствия исходного I_{src} и полученного в результате преобразования I_{dst} изображений можно использовать различные критерии. Считаем, что в нашем случае уместно использовать показатели дисперсии для каждой цветовой компоненты:

$$MSE_R = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [R_{src}(x_i, y_j) - R_{dst}(x_i, y_j)]^2,$$

$$MSE_G = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [G_{src}(x_i, y_j) - G_{dst}(x_i, y_j)]^2,$$

$$MSE_B = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N [B_{src}(x_i, y_j) - B_{dst}(x_i, y_j)]^2,$$

Но наиболее наглядно степень изменения цветовой компонент характеризуют попарно представленные гистограммы распределения пикселей по яркости. Далее опишем работу других алгоритмов цветовой коррекции, а после этого проведем сопоставительный анализ их качества количественно и визуально.

Алгоритм «идеального отражателя». Он основан на предположении о том, что наиболее яркие области изображения относятся к бликам на поверхностях, модель отражения которых такова, что цвет блика равен цвету освещения (дихроматическая модель). Для такой модели по трем цветовым компонентам $R_{src}, G_{src}, B_{src}$ исходного изображения I_{src} необходимо определить максимальные значения яркостей пикселей $R_{src \max}, G_{src \max}, B_{src \max}$. После этого проводится масштабирование яркости пикселей изображения I_{src} по правилу

$$R_{dst} = \frac{R_{src} \cdot 255}{R_{src \max}}, \quad G_{dst} = \frac{G_{src} \cdot 255}{G_{src \max}}, \quad B_{dst} = \frac{B_{src} \cdot 255}{B_{src \max}}.$$

Алгоритм растяжения контрастности «autolevels» основан на идее растяжения интенсивности цветовой компонент $R_{src}, G_{src}, B_{src}$ исходного

изображения на весь диапазон. Для этого определяются максимумы и минимумы цветовой компонент $R_{src \max}, G_{src \max}, B_{src \max}, R_{src \min}, G_{src \min}, B_{src \min}$, а затем проводятся следующие преобразования:

$$R_{dst} = (R_{src} - R_{src \min}) \cdot \frac{(225 - 0)}{(R_{src \max} - R_{src \min})},$$

$$G_{dst} = (G_{src} - G_{src \min}) \cdot \frac{(225 - 0)}{(G_{src \max} - G_{src \min})},$$

$$B_{dst} = (B_{src} - B_{src \min}) \cdot \frac{(225 - 0)}{(B_{src \max} - B_{src \min})}.$$

Алгоритм повышения контрастности цветных изображений (Equalization) с переходом из цветового RGB-пространства в пространство YUV . Это цветовая модель, в которой цвет представляется как 3 компоненты – яркость (Y) и две цветоразностных (U и V). В изображении в формате YUV эквализации подвергается только компонента яркости (Y) с помощью функции из библиотеки OpenCV

```
img_yuv[:, :, 0] =
cv2.equalizeHist(img_yuv[:, :, 0]).
```

Затем производится обратное преобразование изображения из формата YUV в формат RGB. При этом баланс цвета сохраняется без изменений, так как цветоразностные компоненты U и V преобразования не подвергались.

```
import cv2
import numpy as np
img = cv2.imread('input.jpg')
img_yuv =
cv2.cvtColor(img, cv2.COLOR_BGR2YUV)
# equalize the histogram of the Y channel
img_yuv[:, :, 0] =
cv2.equalizeHist(img_yuv[:, :, 0])
# convert the YUV image back to RGB
img_output =
cv2.cvtColor(img_yuv, cv2.COLOR_YUV2BGR)
cv2.imshow('Color input image', img)
cv2.imshow('Histogram equal-
ized', img_output)
cv2.waitKey(0)
```

Для наглядности приведем пример программного кода алгоритма повышения контрастности цветного изображения, синтезированного на языке Python с использованием функций OpenCV.

Возможны и другие варианты построения таких алгоритмов. Однако следует подчеркнуть, что выбор конкретного алгоритма цветовой коррекции камеры (или их комбинации) обусловлен особенностями освещения сцены. Поэтому целесообразно при создании программного обеспечения этой задачи обеспечить возможность смены алгоритма.

Стенд для совместной балансировки цвета камер стереопары

Для выполнения совместной цветовой коррекции стереопары состоящей из двух Web-камер использовался лабораторный стенд, общий вид которого показан на рис. 1, а. Используется компьютер с

программным обеспечением, стереопара на базе двух идентичных Web-камер с возможностью их подключения к компьютеру через USB-порты, а

также специальный экран на штативе для крепления специальной метки-мишени. На рис. 1, б более детально видны особенности конструкции стереопары.



Рис. 1. Общий вид лабораторного стенда – а; конструкция стереопары – б

При жестком креплении камер относительно друг друга, предусмотрена возможность изменить положение стереопары по высоте, развернуть её на 360° вокруг вертикальной оси. Каждую камеру можно развернуть на небольшой угол в горизонтальной и вертикальной плоскостях. Но главное — регулировка базового расстояния между камерами с использованием специальной контрольной линейки. Это обеспечивает установку базового расстояния с точностью до долей мм.

Структура алгоритма цветовой коррекции web-камер стереопары

Для решения задачи совместной цветовой коррекции камер стереопары авторами предложено использовать алгоритм, основанный на определении вертикального и горизонтального параллакса изображения метки-мишени при совмещении изобра-

жений левой и правой камер Δx и Δy . Полученные смещения изображений Δy используются для ректификации изображений стереопары (image rectification) — приведению опорных точек к единой эпиполярной линии. Затем на левом и правом изображениях сцены выбираются два фрагмента одинаковой размерности. Назовем их областями интереса ROI (Region Of Interest). На основании полученных ранее смещений Δx и Δy проводится цифровая конвергенция (совмещение с точностью до пикселя) этих фрагментов (ROI_left и ROI_right). Далее производится коррекция цветовых характеристик правого канала относительно левого, который выбирается в качестве опорного.

Далее приведено более подробное описание принципов работы алгоритма, а его обобщенная структурная схема показана на рис. 2.

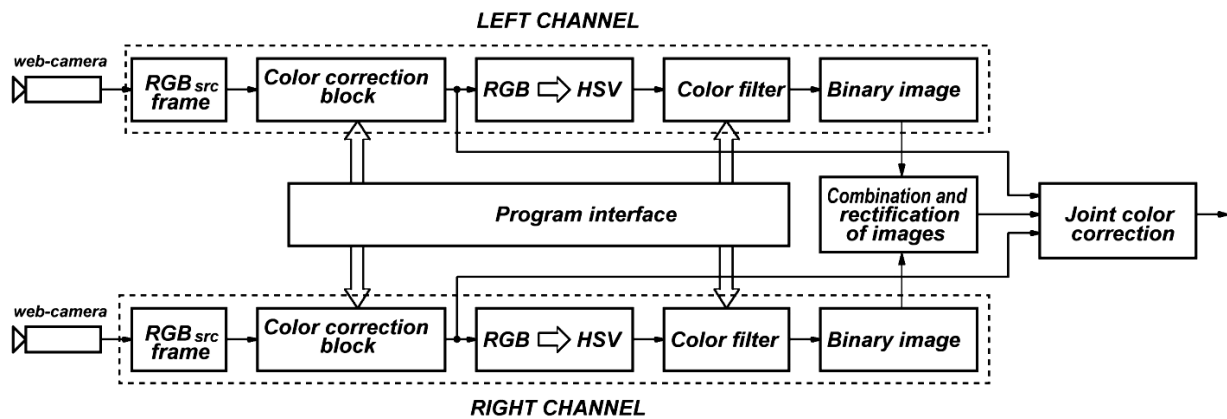


Рис. 2. Структурная схема алгоритма цветовой коррекции камер стереопары

При решении различных задач стереосопоставления (измерении дальности и построении карт глубины сцены) принято пользоваться опорными точками, обладающими определенными отличительными признаками (яркость, цвет и пр.). Авторы считают, что наиболее устойчивой и мало чувствительной к изменениям освещенности сцены является метка-мишень насыщенного цвета (в нашем случае красного). Она закреплена на специальном экране. Поэтому для обнаружения и надежного детектирования опорной точки сцены в кадре использована методика её выделения по цветовому признаку.

Будем полагать, что съёмка сцены с размещенной на ней цветной меткой-мишенью производится стереопарой, предварительно откалиброванной по стандартной методике.

Принятые сигналы web-камер являются последовательностью изображений представленных в цветовом пространстве RGB. В связи с высокой изменчивостью освещенности сцены обычно необходима предварительная цветовая коррекция и балансировка белого независимо для каждой камеры. Методика этих преобразований нами уже обсуждалась достаточно подробно. Конкретный алгоритм цвето-

вой коррекции можно выбрать интерактивно в соответствующем меню интерфейса программы.

Использование цветового пространства RGB малопродуктивно, так как поиск областей нужного цвета предполагает анализ всех трех составляющих R, G и B. Поэтому необходимо перейти в другое цветовое пространство.

Для поиска на изображении объектов по цвету и яркости, как правило, применяется цветовое пространство HSV, где Hue, Saturation и Value – параметры, определяющие цветовой тон, насыщенность и яркость. Они нормированы следующим образом: Hue – 0 – 360°; Saturation – (0 – 100)%; Value – (0 – 100)%. Поэтому наиболее важной процедурой после ввода и цветовой коррекции видеоданных стереоканалов было преобразование цветового пространства RGB в пространство HSV с помощью соответствующей функции OpenCV:

```
hsv=cv2.cvtColor(frame,cv2.COLOR_BGR2HSV)
```

Существенное влияние на качество обнаружения оказывает точность выбора компонент, соответствующих цвету наблюдаемого объекта при различных условиях освещения сцены. Любая точка на шкале H определяет двухмерную область с разными значениями S и V. Но для выбранного значения H целевая область в этом двухмерном пространстве надежнее определяется условием: $V > V_{\min}$ и $S > S_{\min}$, где V_{\min} , S_{\min} – некоторые константы. Поэтому принято выбирать целевой диапазон на шкале H, указывая два значения – H_{\min} и H_{\max} .

Отметим, что диапазон значений параметров HSV для выделения красного цвета обычно лежит в пределах

```
lower_range = np.array([0,50,100])
up_range = np.array([10,255,255]).
```

Процедуру выбора рабочего диапазона параметров пространства HSV для наблюдаемого объекта лучше осуществлять интерактивно в зависимости от текущих условий освещения и цвета самого объекта. Такой подход реализован в рассматриваемом далее алгоритме с помощью элементов настройки (движков) интерфейса. После выбора диапазона нужных цветовых компонент (например, синего цвета) производилась фильтрация (настройка цветовой маски) с помощью пороговой функции

```
mask_blue = cv2.inRange(hsv,
low_range,up_range).
```

После цветовой фильтрации изображение кадра представляется в оттенках серого цвета (диапазон значений яркости пикселей от 0 до 255). Такая процедура позволяет надежно детектировать объект по его цвету, однако не позволяет полностью подавить помехи фона, содержащие похожие цветовые компоненты. Эту трудность можно легко преодолеть путем бинаризации с отсечением по порогу яркости отфильтрованного по цвету изображения. Для этого использовалась функция

```
mask_blue = cv2.threshold(mask_blue,p,
255,cv2.THRESH_BINARY) [1]
```

Обратите внимание на выбор порога отсечения по яркости p в этой функции, зависящий от уровня помех. Его желательно устанавливать интерактивно.

Следующим шагом при построении алгоритма является определение координат метки-мишени в текущих кадрах левой и правой камер. При работе с бинарными изображениями наиболее продуктивным способом этого решения является вычисление моментных инвариантов такого изображения, которые позволяют определить координаты с высокой степенью точности. Моменты изображения вычисляются с помощью функции

```
moments = cv2.moments(thresh,1).
```

Функция moments возвращает массив моментов вплоть до третьего порядка. Однако для вычисления координат центра объекта требуются только моменты первого порядка m01 и m10, а также момент нулевого порядка m00. Они определяются таким образом:

```
dM01 = moments['m01'],
dM10 = moments['m10'],
dArea = moments['m00'].
```

Момент m00 – это количество всех единичных пикселей принадлежащих выделенному объекту, а моменты m01 и m10 – суммы X и Y координат этих пикселей. Для определения координат центра объекта необходимо выполнить нормировку этих моментов на момент нулевого порядка.

При проведении этой процедуры целесообразно выполнить дополнительную пороговую операцию, позволяющую отфильтровать ложные объекты, вероятность появления которых при работе цветowego фильтра не исключается. В этом случае при наличии априорной информации о размерах наблюдаемого объекта можно по условию if dArea > N, где N – количество единичных пикселей момента m00, устранить ложные объекты. В приведенном далее примере программа будет реагировать только на моменты, содержащие больше 50 пикселей

```
if dArea > 50:
x = int(dM10 / dArea),
y = int(dM01 / dArea).
```

В этом примере устраняются случайные засветки красного цвета, имеющие сравнительно малую площадь в кадре.

После бинаризации изображений и определения координат метки-мишени в левом и правом каналах эти изображения объединяют методом прямого наложения. По результатам совмещения этих изображений вычисляются значения горизонтального и вертикального параллакса Δx и Δy , которые используются как для ректификации изображений стереопары, так и для цифровой конвергенции выделяемых областей ROI.

На рис. 3 показана структурная схема одного из возможных вариантов реализации алгоритмов цветовой коррекции web-камеры одного из каналов. В данном случае можно использовать либо алгоритм «серого мира», наиболее подходящий при существенной разбалансировке белого цвета изображений, либо алгоритм «идеального отражателя» при наличии бликов в кадре. Эти и другие алгоритмы цвето-

вой коррекции можно использовать и в различной комбинации.

Далее рассмотрим работу алгоритма совместной цветовой коррекции изображений сцены в левом и правом стереоканалах (joint color correction). Его структурная схема приведена на рис. 4. После ректификации изображений, выполняемой путем

сдвига изображения правого канала относительно левого на величину вертикального параллакса Δy , производится выделение областей интереса (ROI_left и ROI_right). Отметим, что конвергенция (точное совмещение) этих изображений производится с учетом горизонтального параллакса Δx между левым и правым каналом стереопары.

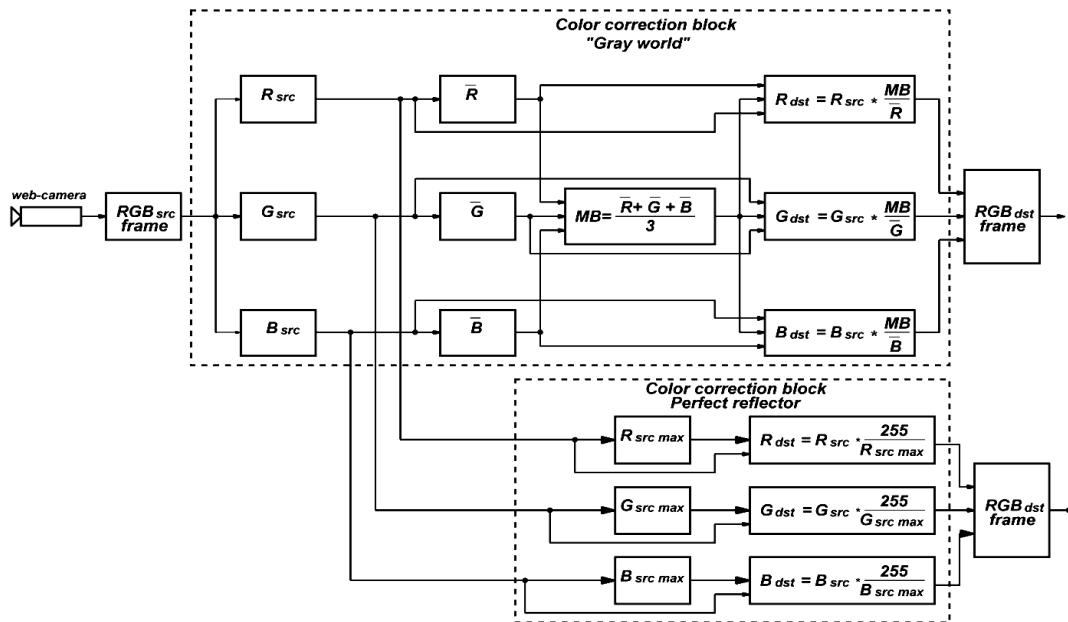


Рис. 3. Структурная схема возможных алгоритмов цветовой коррекции отдельного канала стереопары

Следует помнить, что размеры ROI каналов должны адекватно отображать характер освещенности сцены. Но их нельзя выбирать и слишком большими, поскольку это приведет к неоправданному увеличению вычислительной сложности алгоритма. Оптимальным, на наш взгляд, размером ROI является центральный фрагмент изображения 100x100 пикселей при формате кадров каналов 480x640. Несомненно, что выбирать фрагменты ROI пользователю следует интерактивно.

Совместная цветовая коррекция камер стереопары проводится в предположении, что одно из изображений является базовым. В нашем случае принимаем за эталон баланс белого и яркость изображения ROI_left. Второе изображение (ROI_right) будем считать зависимым, и корректировать компоненты цвета этого изображения относительно базового.

При сопоставлении ROI можно построить гистограммы распределения цветových компонент R, G и B для визуального сравнения. Однако ясно, что такой подход малоэффективен. Гораздо продуктивнее оценивать и сопоставлять числовые интегральные характеристики отдельных цветových компонент ROI_left и ROI_right: Поэтому в этом алгоритме вычисляются суммарные показатели яркости отдельных цветových компонент для левого и правого каналов:

$$R_{\Sigma}(ROI_{left}) = \sum_{k=1}^K \sum_{l=1}^L R_{src\ left}(x_k, y_l),$$

$$G_{\Sigma}(ROI_{left}) = \sum_{k=1}^K \sum_{l=1}^L G_{src\ left}(x_k, y_l),$$

$$B_{\Sigma}(ROI_{left}) = \sum_{k=1}^K \sum_{l=1}^L B_{src\ left}(x_k, y_l),$$

где x_k, y_l – текущие номера строк и столбцов исходного изображения ROI_{left} . Аналогичным образом рассчитываются показатели $R_{\Sigma}(ROI_{right})$, $G_{\Sigma}(ROI_{right})$ и $B_{\Sigma}(ROI_{right})$ для правого канала.

На основании информации о суммарной яркости цветových компонент изображений ROI_{left} и ROI_{right} вычисляются поправочные коэффициенты:

$$k_R = \frac{R_{\Sigma}(ROI_{left})}{R_{\Sigma}(ROI_{right})}, \quad k_G = \frac{G_{\Sigma}(ROI_{left})}{G_{\Sigma}(ROI_{right})},$$

$$k_B = \frac{B_{\Sigma}(ROI_{left})}{B_{\Sigma}(ROI_{right})}.$$

Полученные коэффициенты k_R, k_G и k_B используются для корректировки баланса белого зависимого изображения (в нашем случае правого стереоканала) относительно базового (левого стереоканала). Цветовые компоненты R, G и B умножаются на эти коэффициенты.

Отметим, что при достаточно точной балансировке белого обычно отпадает необходимость корректировки яркости в стереоканалах.

Тем не менее, рассмотрим методику компенсации различий интегральной яркости левого и правого каналов.

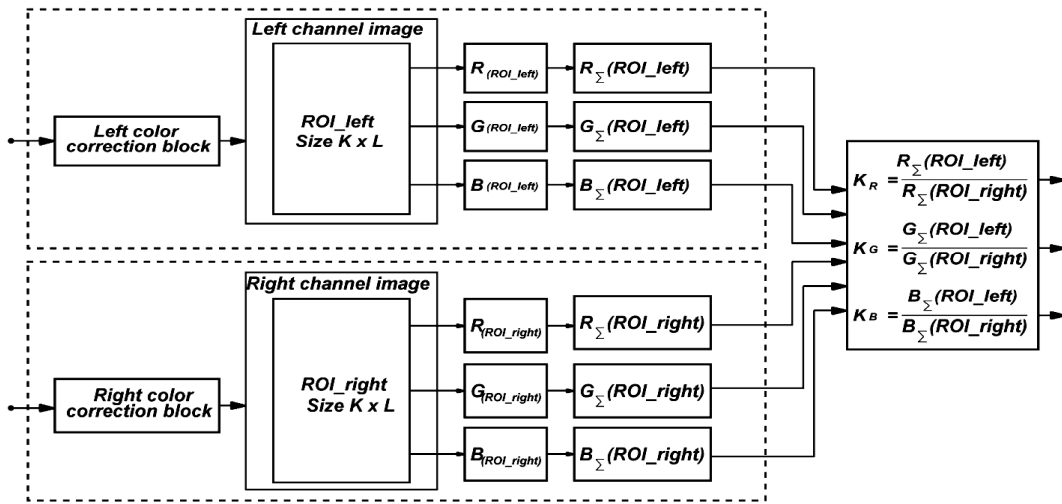


Рис. 4. Структурная схема корректировки баланса белого web-камер стереопары

Для комфортного восприятия изображений при воспроизведении стереоскопической записи необходимо скорректировать суммарную яркость базового изображения в области интереса $I_{\Sigma}(ROI_{right})$. Её можно вычислить, используя весовые коэффициенты p , по формуле:

$$I_{\Sigma}(ROI_{right}) = p_1 R_{\Sigma}(ROI_{left}) + p_1 G_{\Sigma}(ROI_{left}) + p_1 B_{\Sigma}(ROI_{left}).$$

Значения коэффициентов p определяются физиологическими свойствами человеческого глаза [16] и принимаются равными

$$p_1 = 0.2126; \quad p_2 = 0.7152; \quad p_3 = 0.2126.$$

Приведем подробную запись формулы для расчёта интегральной яркости базового (левого) канала:

$$I_{\Sigma}(C) = p_1 \sum_{k=1}^K \sum_{l=1}^L R_{src\ left}(x_k, y_l) + p_2 \sum_{k=1}^K \sum_{l=1}^L G_{src\ left}(x_k, y_l) + p_3 \sum_{k=1}^K \sum_{l=1}^L B_{src\ left}(x_k, y_l).$$

Аналогичным образом вычисляется показатель суммарной яркости $I_{\Sigma}(ROI_{right})$ и для правого (зависимого) канала. Но для достижения в стереоканалах баланса суммарных яркостей, необходимо полученные для этого канала показатели корректировать с помощью поправочных коэффициентов k_R, k_G, k_B .

$$I_{\Sigma}(ROI_{left})^* = \left[p_1 \sum_{k=1}^K \sum_{l=1}^L R_{src\ right}(x_k, y_l) \right] \cdot k_R + \left[p_2 \sum_{k=1}^K \sum_{l=1}^L G_{src\ right}(x_k, y_l) \right] \cdot k_G + \left[p_3 \sum_{k=1}^K \sum_{l=1}^L B_{src\ right}(x_k, y_l) \right] \cdot k_B.$$

Таким образом можно скорректировать относительное отклонение суммарной яркости каналов после корректировки баланса белого.

Методы проведения экспериментальных исследований и основные результаты

Для оценки эффективности предлагаемых методов и алгоритмов цветовой коррекции Web-камер стереопары были проведены экспериментальные исследования, которые можно разделить на две группы – анализ качества алгоритмов для цветовой коррекции отдельного стереоканала (Channel color correction block), и оценка качества работы алгоритмов совместной цветовой коррекции (Joint color correction block). С помощью лабораторного стенда (рис. 1) и программного обеспечения, структура которого показана на рис. 2, проводилась стереоскопическая съёмка сцены с меткой-мишенью на ней (рис. 5).

На рис. 5 показаны левый и правый кадры видеопотока стереопары с указанием координат метки-мишени. Важно отметить, что программа позволяет вести обработку данных в реальном времени. Кроме этого, предусмотрена возможность записи входной видеопоследовательности нужной продолжительности. Эта процедура выполняется интерактивно. В системе предусмотрена возможность ввода данных как со стереопары в режиме online, так и из сохраненного ранее файла. Последний режим позволяет выполнять обработку данных многократно, используя различные алгоритмы. При этом появляется возможность сопоставить эффективность работы различных алгоритмов на одном и том же массиве исходных данных. Также интерактивно можно сделать копию фрейма левого или правого каналов по нажатию комбинации клавиш клавиатуры Alt + PrtScn. Это изображение в программе используется для экспресс-тестирования условий освещения сцены и выбора соответствующего алгоритма цветовой коррекции канала в этот период работы.

Результаты обработки видеопоследовательности стереопары после бинаризации кадров и оценки координат метки-мишени показаны на рис. 6. Совмещение каналов методом прямого наложения увеличивает наглядность при оценке параллакса Δx и Δy , и афиширует их значения (рис. 6, с).

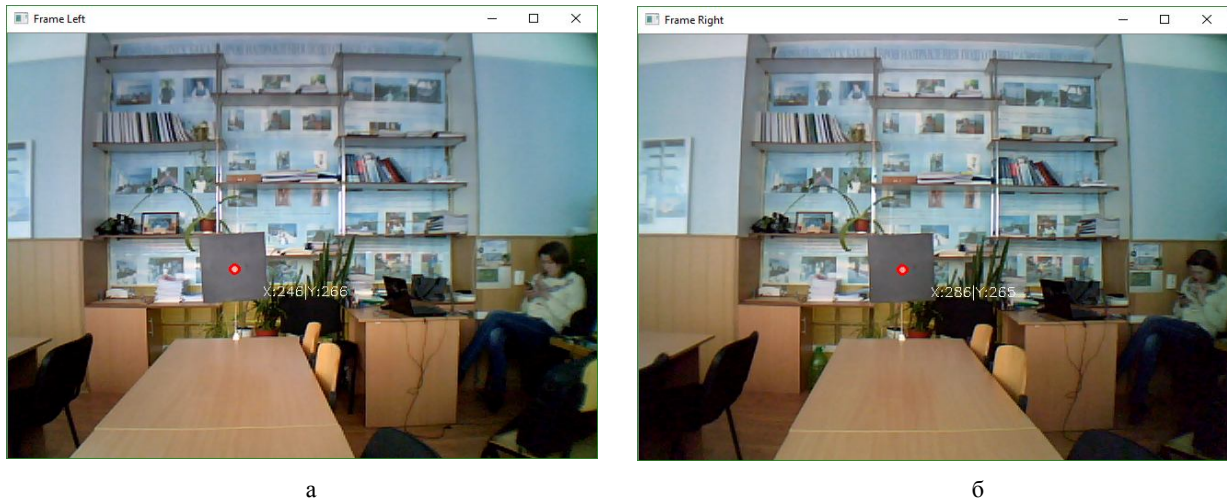


Рис. 5. Фрагмент стереоскопической видеосъемки: а – левый канал, б – правый канал



Рис. 6. Координаты метки-мишени а – в левом канале; б – в правом канале; с – параллаксы Δx и Δy

Рассмотрим далее методику определения эффективности работы алгоритмов цветовой коррекции отдельных каналов. Она складывается из трёх компонент:

- визуальная оценка качества обработки исходного изображения I_{src} . К сожалению, она зачастую носит субъективный характер и зависит от качества зрения экспертов. Это не всегда даёт возможность корректно выделить отличия выходного изображения I_{dst} от входного I_{src} визуально.
- построение гистограмм в цветовом пространстве RGB как функции распределения количества пикселей по градациям яркостей изображения в цветовом диапазоне от 0 до 255. Гистограммы удобно строить попарно в одном окне (для каждой из трех цветовых компонент). В этих окнах гистограммы входных изображений I_{src} отображаются сплошными кривыми, а гистограммы выходных изображений I_{dst} – пунктирными. По ним наглядно видно, как в результате действия тестируемого алгоритма изменяется соотношение ярких и темных пикселей (рис. 8).
- количественные показатели цветовой коррекции.

Для этого использован специальный программный блок, позволяющий оценить эффективность работы алгоритмов цветовой коррекции левого или правого стереоканала по отдельному кадру для текущих условий освещенности сцены, или по какому-нибудь другому изображению.

В качестве обобщенного показателя эффективности работы используемых алгоритмов авторами предложен статистический вектор рассогласования яркости σ_{Σ} исходного изображения I_{src} и преобразованного I_{dst} . Далее будем именовать его как «Вектор цветового баланса» или (англ. – Color balance vector). Он представлен в ортогональном трехмерном цветовом пространстве RGB. Цветовое пространство в этом случае принято представлять в виде RGB цветового куба, показанного на рис. 7. Известно, что точки диагонали этого куба, соединяющей две противоположные вершины с координатами (0,0,0) и (255,255,255), получили название шкалы серого цвета (Gray scale). Они меняют свой оттенок от черного до ярко белого.

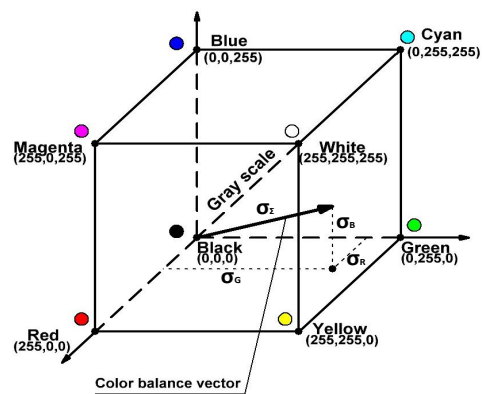


Рис. 7. Вектор цветового баланса в ортогональном пространстве RGB

В этой системе координат начало вектора σ_{Σ} совмещено с нулевой (черной) точкой, его модуль определяется совокупностью показателей дисперсий для каждой цветовой компоненты MSE_R , MSE_G , и MSE_B , используемых для описания степени соответствия исходных I_{src} и полученных в результате преобразования I_{dst} изображений. Формулы для вычисления этих дисперсий приведены выше. Величина модуля «вектора цветового баланса» равна

$$\sigma_{\Sigma} = \sqrt{MSE_R + MSE_G + MSE_B}$$

или определяется по сумме квадратов отклонений (Standard deviation $\sigma_{R,G,B}$) следующим образом:

$$\sigma_{\Sigma} = \sqrt{(\sigma_R^2 + \sigma_G^2 + \sigma_B^2)}$$

При необходимости можно анализировать и угловое положение вектора цветового баланса σ_{Σ} определяя эти углы по отношению модуля вектора к его проекциям на оси координат куба $(\sigma_R, \sigma_G, \sigma_B)$.

Результаты экспериментальных исследований приведены далее в виде табл. 1–5, каждая из которых содержит данные анализа эффективности одного из алгоритмов цветовой коррекции. В ячейки таблицы включены исходное изображение I_{src} и преобразованное I_{dst} , данные статистической обработки и гистограммы распределения яркостей трех цветовых компонент. Гистограммы входного изображения расположены попарно для каждой цветовой компоненты. Это создает удобство для анализа. Также по результатам статистического анализа рассчитывается и вносится в таблицу модуль вектора цветового баланса σ_{Σ} . Анализ отдельного из исследуемых алгоритмов и сопоставительный анализ их качества необходимо проводить на основании объективного комплексного критерия, который должен учитывать как характер визуального восприятия изменений изображения, так и варибельность гистограмм в результате преобразований. Не менее важным аспектом анализа является и учет статистических показателей. Необходимо также оценивать модуль и угловое положение вектора цветового баланса σ_{Σ} в ортогональном цветовом пространстве RGB.

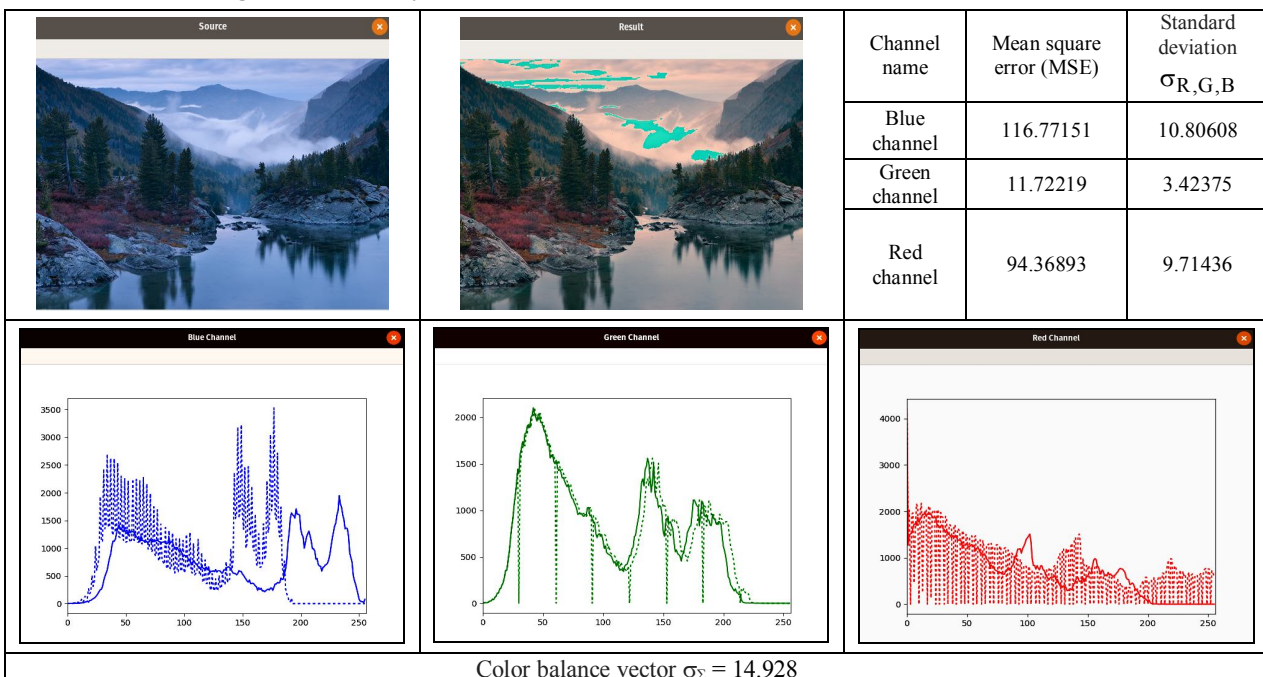
Сопоставительный анализ четырех алгоритмов цветовой коррекции показал, что каждый из них обладает как достоинствами, так и недостатками. Следует обратить внимание, что при обработке изображения алгоритмом «Gray World» произошло существенное нарушение цветового баланса (табл.1). Это означает, что такой алгоритм эффективен не при всех условиях освещения сцены. Пример, приведенный в табл. 5, показывает, что при определенных условиях он вполне применим и достаточно эффективен.

Описанная методика оценки эффективности цветовой коррекции в полной мере применима и для совместной цветовой балансировки камер стереопары. Понятно, что при отсутствии заметных нарушений в изображении баланса белого расхождение гистограмм исходного и обработанного изображений уменьшается, и в пределе становится нулевым. Следовательно, позитивным фактором является и уменьшение модуля вектора цветового баланса.

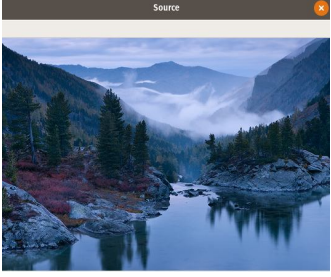
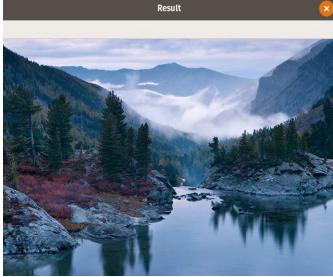
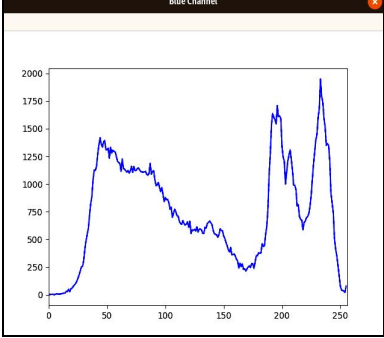
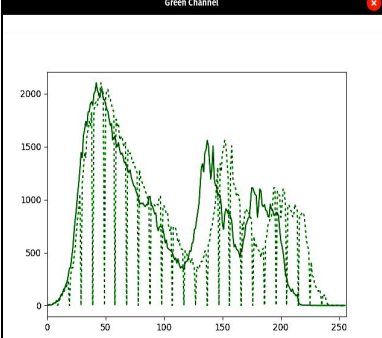
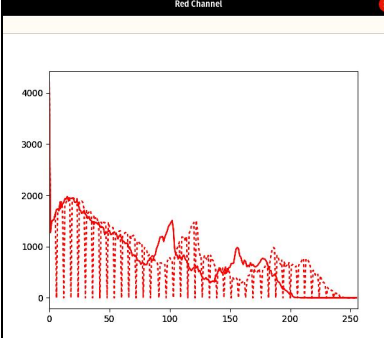
Для удобства анализа данные о величине модуля вектора цветового баланса σ_{Σ} сведены в табл. 6. По мнению авторов практически наиболее предпочтительно использование алгоритмов «Autolevels» и «Equalization». Последний наиболее эффективен при малых уровнях освещения сцены, так как позволяет показатели яркости изображения привести к среднему уровню освещенности.

По мнению авторов практически наиболее предпочтительно использование алгоритмов «Autolevels» и «Equalization». Последний наиболее эффективен при малых уровнях освещения сцены, так как позволяет показатели яркости изображения привести к среднему уровню освещенности.

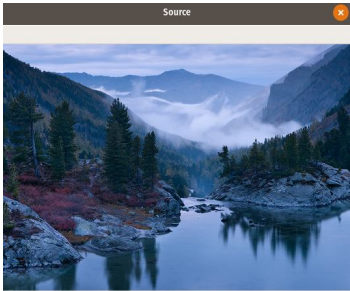
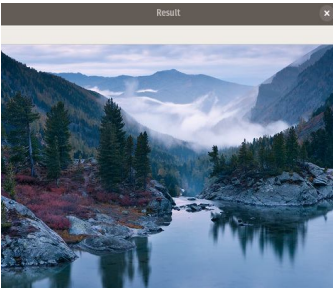
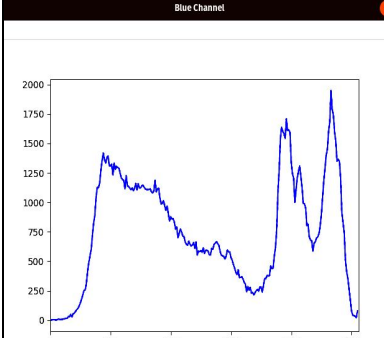
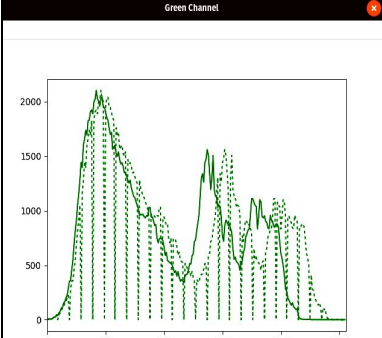
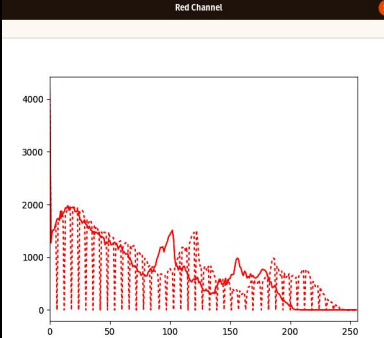
Таблица 1 – Processing method – «Gray World»



Таблиця 2 – Processing method – «Perfect Reflector»

| | | | | | |
|---|---|---|---|-------------------------------------|--|
|  |  | Channel name | Mean square error (MSE) | Standard deviation $\sigma_{R,G,B}$ | |
| | | Blue channel | 2.15407 | 1.46767 | |
| | | Green channel | 83.04019 | 9.11263 | |
| | | Red channel | 74.05681 | 8.60562 | |
|  |  |  | Color balance vector $\sigma_{\Sigma} = 12.619$ | | |

Таблиця 3 – Processing method – «Autolevels»

| | | | | | |
|---|---|---|---|-------------------------------------|--|
|  |  | Channel name | Mean square error (MSE) | Standard deviation $\sigma_{R,G,B}$ | |
| | | Blue channel | 0.00000 | 0.00000 | |
| | | Green channel | 80.04215 | 8.94662 | |
| | | Red channel | 76.05679 | 8.72105 | |
|  |  |  | Color balance vector $\sigma_{\Sigma} = 12.494$ | | |

Таблиця 6 – Модуль вектора цветового балансу

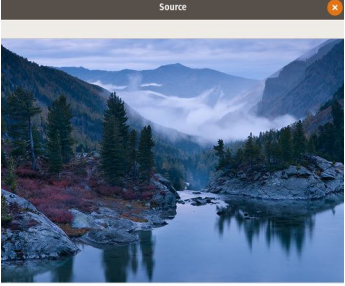
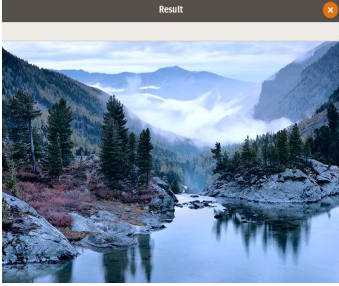
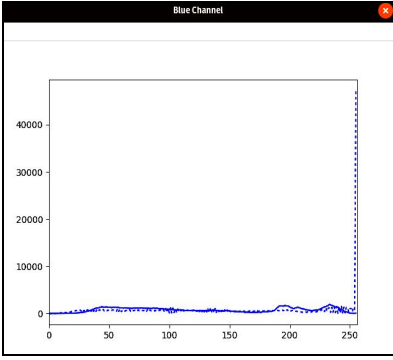
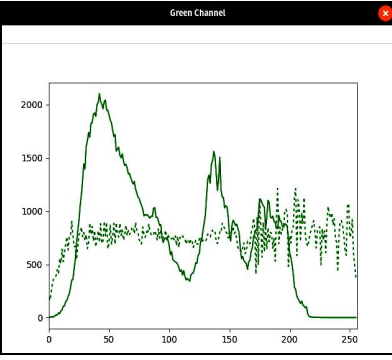
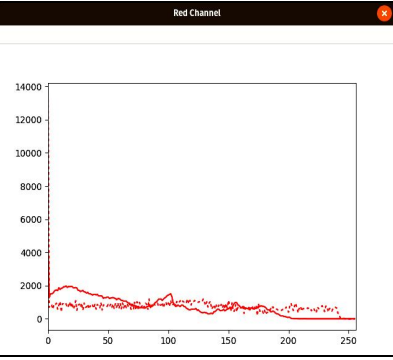
| Processing method | Color balance vector σ_{Σ} |
|-------------------|--|
| Gray World | 14.928 |
| Perfect Reflector | 12.619 |
| Autolevels | 12.494 |
| Equalization | 17.985 |

Заключення

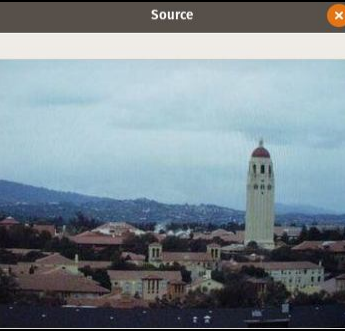
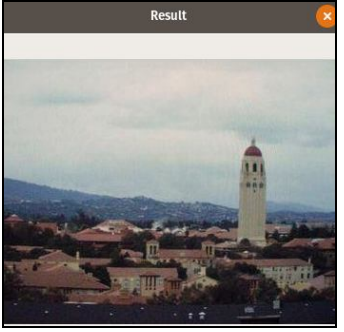
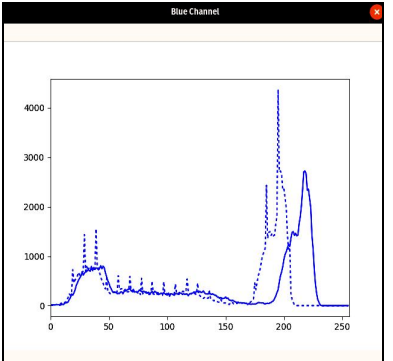
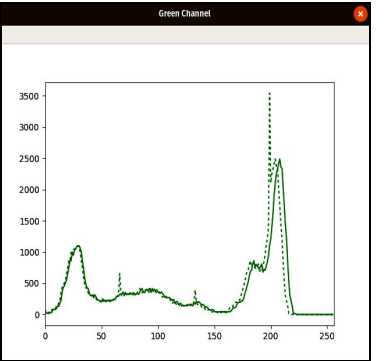
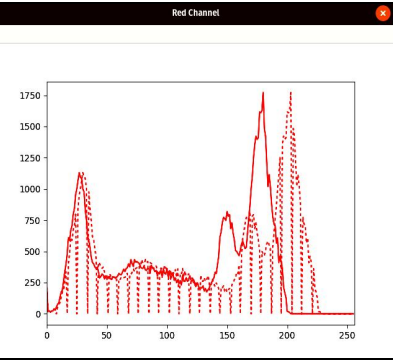
Предложена новая методика и алгоритмы цветовой коррекции web-камер стереопары в системах бинокулярного зрения. Экспериментально проана-

лизирована эффективность их использования. Создано программное обеспечение на языке Python с использованием ресурсов библиотеки OpenCV, позволяющее реализовать работу этих алгоритмов в реальном времени. Предусмотрена возможность адаптации работы программы к различным изменениям условий освещения сцены. Использование этих результатов дает возможность повысить качество работы систем бинокулярного зрения при решении различных задач как стереовизуализации, так и стереосопоставления.

Таблиця 4 – Processing method – «Equalization»

| | | | | | |
|---|---|---|--|-------------------------------------|--|
|  |  | Channel name | Mean square error (MSE) | Standard deviation $\sigma_{R,G,B}$ | |
| | | Blue channel | 110.50516 | 10.51214 | |
| | | Green channel | 110.10154 | 10.49293 | |
| | | Red channel | 102.85079 | 10.14153 | |
|  |  |  | Color balance vector $\sigma_2 = 17.985$ | | |

Таблиця 5 – Processing method – «Gray World»

| | | | | | |
|---|---|---|--|-------------------------------------|--|
|  |  | Channel name | Mean square error (MSE) | Standard deviation $\sigma_{R,G,B}$ | |
| | | Blue channel | 110.11258 | 10.49345 | |
| | | Green channel | 7.71392 | 2.77739 | |
| | | Red channel | 82.72612 | 9.09539 | |
|  |  |  | Color balance vector $\sigma_2 = 14.161$ | | |

СПИСОК ЛІТЕРАТУРИ

1. Использование web-камер в качестве источника потока стереопар / С. И. Протасов, С. Д. Кургалин, А. А. Крыловецкий // Воронеж, Вестник ВГУ, Серия: системный анализ и информационные технологии, 2011, № 2
2. Цифровая обработка изображений как сигналов. Компьютерная графика, лекция, [Электронный ресурс]. – Режим доступа <http://www.myshared.ru/slide/529382/>
3. Методы цветокоррекции стереовидео / В. Людвиченко // Video Group CS MSU Graphics & Media Lab (Video Group), [Электронный ресурс]. – Режим доступа www.compression.ru/video/
4. Калибровка камеры в программе Camera Calibration Toolbox for Matlab, [Электронный ресурс]. – Режим доступа http://www.vision.caltech.edu/bouguetj/calib_doc/
5. Linda G. Shapiro and George C. Stockman (2001). Computer Vision. Prentice Hall. pp. 580.

6. Стереореко́нструкція / А. Конушин [и др.] // МГУ, ВМиК. – 2008. [Электронный ресурс]. – Режим доступа <http://courses.graphicon.ru/main/vision2008>.
7. Функции стереозрения в OpenCV / А. Кручинин, [Электронный ресурс]. – Режим доступа <https://docplayer.ru/53282398-Funkcii-stereozeniya-v-opencv.html>
8. Методы формирования изображений стереопары с заданным значением параллакса / В. Г. Чафонова, И. В. Газеева, Научно-технический вестник информационных технологий, механики и оптики Scientific and Technical Journal of Information Technologies, Mechanics and Optics 2014, № 6 (94), Санкт-Петербургский государственный университет кино и телевидения, Санкт-Петербург, 191119, Российская Федерация, vi777@nextmail.ru
9. Joseph Howse, Joe Minichino, Learning OpenCV 3 Computer Vision with Python - Second Edition, Packt Publishing, September 2015, Packt Publishing, ISBN: 9781785289774
10. Saurabh Kapur, Computer Vision with Python 3, Packt Publishing, August 2017, ISBN: 978-1-78829-976-3.
11. Prateek Joshi, OpenCV with Python By Example, Packt Publishing, September 2015, ISBN: 978-1-78528-393-2..
12. Библиотека компьютерного зрения OpenCV [Электронный ресурс]. – Режим доступа http://docs.opencv.org/trunk/doc/py_tutorial/py_objdetect/py_face_detection/py_face_detection.html.
13. Разработка мультимедийных приложений с использованием библиотек OpenCV и IPP [Электронный ресурс] / А. В. Бовыкин [и др.] – Электрон. текстовые данные. – М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. – 515 с. – Режим доступа: <http://www.iprbooksshop.ru/39564/>
14. OpenCV на python: поиск цветного объекта / О. Евсегнеев, [Электронный ресурс]. – Режим доступа <http://robotclass.ru/tutorials/opencv-moments-color-object-search/>
15. Гистограмма рисования в OpenCV with Python, [Электронный ресурс]. – Режим доступа <http://qaru.site/questions/371882/drawing-histogram-in-opencv-python>
16. Хьюбел Д. Глаз, мозг, зрение / под ред. А. Л. Бызова. // М.: Мир, 1990. – 172 с

Рецензент: д-р техн. наук, проф. С. В. Козелков,
Державний університет телекомунікацій, Київ

Received (Надійшла) 28.02.2019

Accepted for publication (Прийнята до друку) 06.04.2019

Розробка методів і засобів колірної корекції web-камер в системах бінокулярного зору

К. Ю. Дергачов, Л. О. Краснов, О. О. Челядін, О. В. Плахотний

Предмет вивчення. У статті запропоновано нові методи і засоби колірної корекції web-камер в системах стереозору з метою підвищення якості їх роботи. **Метою** є порівняльний аналіз показників якості відомих методів корекції кольору і розробка нової методики і робочих алгоритмів спільної процедури колірної корекції та ректифікації фреймів відеопотоку лівої і правої камер. **Завдання:** Ставилося завдання провести теоретичний аналіз показників якості відомих алгоритмів корекції кольору, розробити нові робочі алгоритми, програмні коди цих алгоритмів написати на мові Python з використанням необхідних функцій OpenCV. Провести експериментальні дослідження цих алгоритмів. Оцінити ефективність роботи стереосистеми в лабораторних умовах, а достовірність отриманих результатів перевірити методами статистичного аналізу. **Використовувані методи:** порівняльний аналіз відомих методів і алгоритмів шляхом статистичного моделювання, синтез нових алгоритмів і оцінка ефективності їх роботи шляхом проведення лабораторних натурних випробувань. **Отримані результати:** проведено порівняльний аналіз ефективності роботи відомих методів колірної корекції камер стереосистеми, запропоновані нові більш ефективні алгоритми для вирішення цього завдання. **Висновки.** Наукова новизна отриманих результатів: створено нові алгоритми корекції колірного балансу web-камер, використовуваних в стереоскопічних системах технічного зору, що відрізняються високою точністю колірної корекції і працюють в реальному масштабі часу з застосуванням функцій бібліотеки OpenCV в програмному середовищі Python.

Ключові слова: стереоскопічні системи технічного зору; колірна корекція зображень відеопотоку web-камер стереосистеми; спільна ректифікація і колірний баланс лівого і правого камер.

The method and tools development for web-cameras color correction in binocular vision systems

K. Dergachov, L. Krasnov, O. Cheliadin, O. Plakhotnyi

Subject of study. The article proposes new methods and tools for color correction of web-cameras in stereo vision systems in order to improve the quality of their work. The goal is a comparative analysis of well-known color correction methods quality indicators and the development of a new methodology and working algorithms for the joint procedure for color correction and rectification of left and right cameras video frames. **Objectives:** The task was to carry out a theoretical analysis of well-known color correction algorithms quality indicators, to develop new working algorithms, write algorithms program code using Python with needed OpenCV functions. Conduct experimental study of these algorithms. Evaluate stereo system performance in a laboratory, and test the reliability of the results obtained using statistical analysis methods. **Methods used:** Comparative analysis of known methods and algorithms by statistical modeling, synthesis of new algorithms and evaluation their work effectiveness by conducting laboratory field tests. **The results obtained:** comparative analysis of the performance of known color correction methods of stereo cameras was carried out, new more efficient algorithms were proposed for solving this problem. **Findings.** Scientific novelty of the results: new algorithms for correcting the color balance of webcams used in stereoscopic vision systems have been created, featuring high color correction accuracy and working in real time using OpenCV library functions in Python software environment.

Keywords: stereoscopic vision systems; color correction of video streaming images of web-cameras of a stereo system; joint rectification and color balance of the left and right chambers.