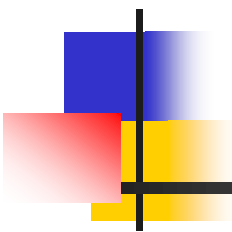


Методы цветокоррекции стереовидео



Виталий Людвиченко

Video Group

CS MSU Graphics & Media Lab



Содержание

- **Введение**
- Методы цветокоррекции
 - Глобальные
 - Локальные
- Обнаружение бликов
- Заключение

Пример проблемного видео

Левый ракурс



Пример проблемного видео

Правый ракурс



Введение

Актуальность цветокоррекции

Проблемы, возникающие в стереовидео:

- Несоответствия цветов между ракурсами:
 - Локальные
 - Глобальные
- Наличие/отсутствие бликов на объектах в разных ракурсах

Причины:

- Различия в матрицах камер
- Различия в светофильтрах
- Разное положение камер относительно источника света

Глобальное несоответствие

Левый ракурс



Глобальное несоответствие

Правый ракурс



Локальное несоответствие

Левый ракурс



Локальное несоответствие

Правый ракурс



Градиентное искажение

Левый ракурс



Градиентное искажение

Правый ракурс



Градиентное искажение

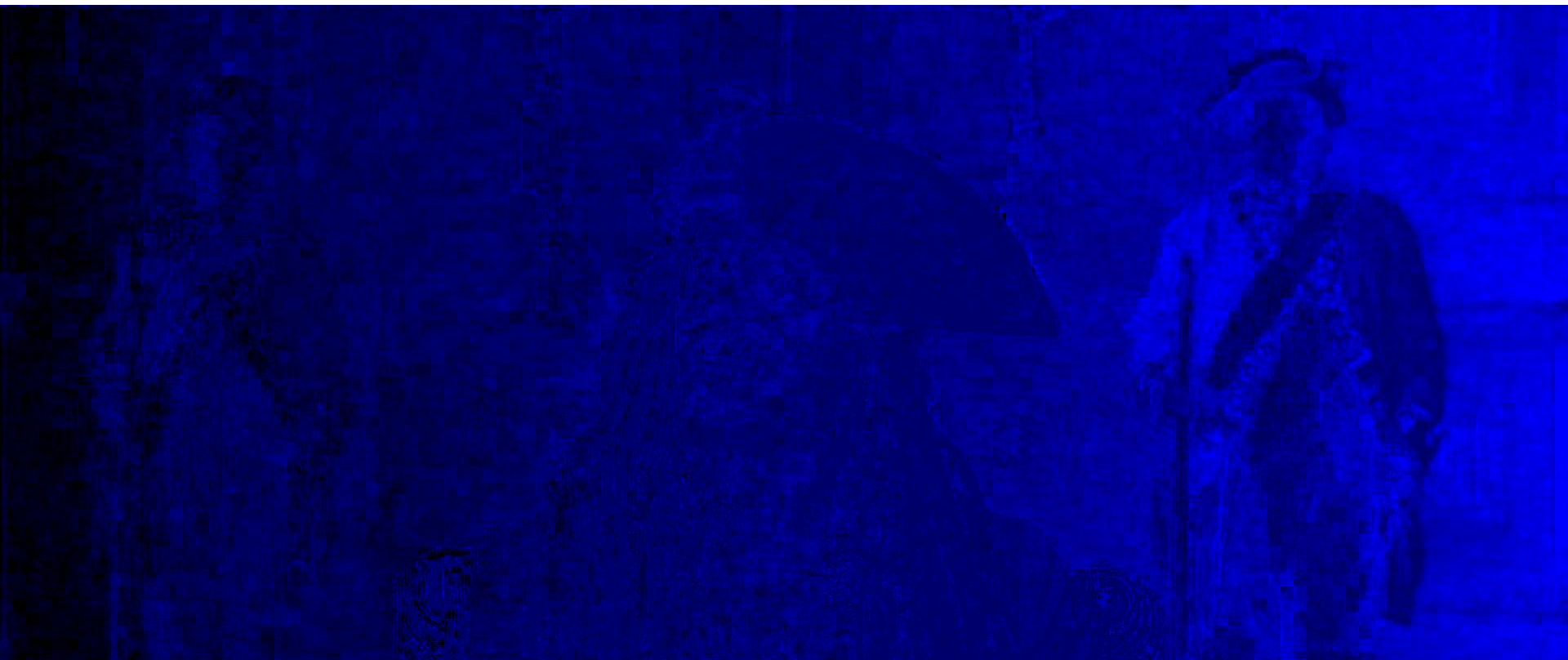
Скомпенсированная разница (1)



Разница между левым и правым скомпенсированным ракурсом

Градиентное искажение

Скомпенсированная разница (2)



Разница между левым и правым скомпенсированным ракурсом по синему каналу



Блики

Левый ракурс





Блики

Правый ракурс





Постановка задачи

Реализовать алгоритм построения трехмерной таблицы преобразования цветов в пространстве RGB, минимизирующей цветовые несоответствия между ракурсами

Дополнительные требования к алгоритму:

- Устойчивость к бликам
- Учитывать пространственное положение пикселя в изображении



Содержание

- Введение
- Методы цветокоррекции
 - Глобальные
 - **Histogram Matching**
 - Global Color Transfer
 - Кластерный метод
 - Локальные
- Обнаружение бликов
- Заключение

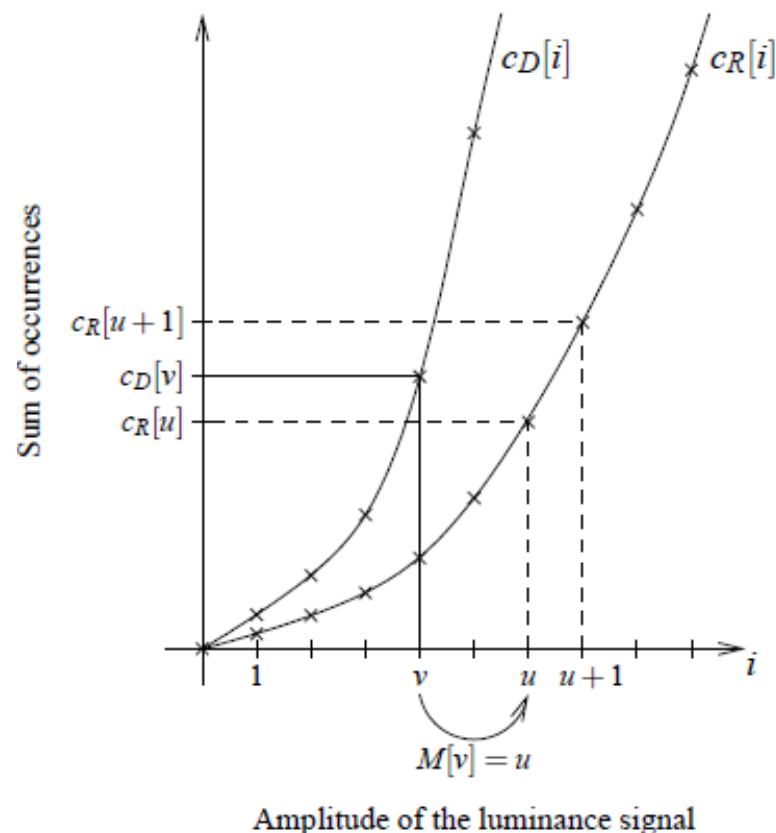
Histogram Matching

- **Задача:**
Найти преобразование гистограммы левого или правого ракурса, чтобы различия между гистограммами ракурсов были минимальны
- **Вход:**
Гистограммы левого и правого ракурса
- **Выход:**
Функция преобразования для гистограммы одного из ракурсов

Histogram Matching

Для каждой компоненты R, G, B:

1. Составляем интегральные (кумулятивные) гистограммы $c_D[v]$ и $c_R[v]$
2. Строим функцию преобразования $M[v] = u$, если $c_R[u] < c_D[v] \leq c_R[u + 1]$
3. Применяем $M[v]$ для каждого пикселя входного изображения



Histogram Matching

Результаты



Левый ракурс

Histogram Matching

Результаты



Правый ракурс

Histogram Matching

Результаты



Скорректированный правый ракурс

Histogram Matching

Результаты



Левый ракурс

Histogram Matching

Выводы

Достоинства:

- Простота реализации
- Низкая вычислительная сложность

Недостатки:

- Работает независимо по каналам
- Неустойчив к локальным несоответствиям



Содержание

- Введение
- Методы цветокоррекции
 - Глобальные
 - Histogram Matching
 - **Global Color Transfer**
 - Кластерный метод
 - Локальные
- Обнаружение бликов
- Заключение

Global Color Transfer

- **Задача:**

Найти линейную функцию преобразования цветового пространства RGB, чтобы целевое изображение имело те же цветовые тона, что и у исходного изображения

- **Вход:**

Исходное и целевое изображения

- **Выход:**

Результирующее изображение

Global Color Transfer

Пример



Исходное изображение



Целевое изображение



Результирующее
изображение

Алгоритм

Вычисление значений

1. Вычисляем средние значения пикселей по каждому из каналов R , G , B для:
 - исходного изображения $(R_{src}, G_{src}, B_{src})$
 - целевого изображения $(R_{tgt}, G_{tgt}, B_{tgt})$
2. Вычисляем матрицу ковариации для:
 - исходного изображения Cov_{src}
 - целевого изображения Cov_{tgt}

Алгоритм

Разложение ковариационных матриц

3. Выполняем сингулярное разложение матриц Cov_{src} и Cov_{tgt} :

$$Cov = U \cdot \Lambda \cdot V^T$$

U и V – ортогональные матрицы, состоящие из собственных векторов, определяющие ориентацию аппроксимирующего эллипсоида

$$\Lambda = \text{diag}(\lambda^R, \lambda^G, \lambda^B)$$

$\lambda^R, \lambda^G, \lambda^B$ – собственные значения, определяющие размеры аппроксимирующего эллипсоида

Алгоритм

Построение преобразования

4. Зная расположения аппроксимирующих эллипсоидов, ищем преобразование как последовательность операций поворота, сдвига и растяжения:

$$I = T_{src} \cdot R_{src} \cdot S_{src} \cdot S_{tgt} \cdot R_{tgt} \cdot T_{tgt} \cdot I_{tgt}$$

$$I = (R, G, B, 1)^T$$

$$I_{tgt} = (R_{tgt}, G_{tgt}, B_{tgt}, 1)^T$$

T_{src} , T_{tgt} , R_{src} , R_{tgt} , S_{src} и S_{tgt} – матрицы сдвига, поворота и растяжения соответственно

Алгоритм

Определение коэффициентов

Коэффициенты, используемые в матрицах преобразования:

$$t_{src}^r = \bar{R}_{src}, t_{src}^g = \bar{G}_{src}, t_{src}^b = \bar{B}_{src}$$

$$s_{src}^r = \lambda_{src}^R, s_{src}^g = \lambda_{src}^G, s_{src}^b = \lambda_{src}^B$$

$$t_{tgt}^r = -\bar{R}_{tgt}, t_{tgt}^g = -\bar{G}_{tgt}, t_{tgt}^b = -\bar{B}_{tgt}$$

$$s_{tgt}^r = 1/\sqrt{\lambda_{tgt}^R}, s_{tgt}^g = 1/\sqrt{\lambda_{tgt}^G}, s_{tgt}^b = 1/\sqrt{\lambda_{tgt}^B}$$

Алгоритм

Матрицы преобразований

5. Определяем матрицы преобразований:

- Матрицы сдвига:

$$T_{src} = \begin{pmatrix} 1 & 0 & 0 & t_{src}^r \\ 0 & 1 & 0 & t_{src}^g \\ 0 & 0 & 1 & t_{src}^b \\ 0 & 0 & 0 & 1 \end{pmatrix}, T_{tgt} = \begin{pmatrix} 1 & 0 & 0 & t_{tgt}^r \\ 0 & 1 & 0 & t_{tgt}^g \\ 0 & 0 & 1 & t_{tgt}^b \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Матрицы поворота: $R_{src} = U_{src}, R_{tgt} = U_{tgt}^{-1}$

- Матрицы масштабирования:

$$S_{src} = \begin{pmatrix} s_{src}^r & 0 & 0 & 0 \\ 0 & s_{src}^g & 0 & 0 \\ 0 & 0 & s_{src}^b & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, S_{tgt} = \begin{pmatrix} s_{tgt}^r & 0 & 0 & 0 \\ 0 & s_{tgt}^g & 0 & 0 \\ 0 & 0 & s_{tgt}^b & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Global Color Transfer

Результаты (1)



Исходное изображение



Целевое изображение



Результирующее
изображение

Global Color Transfer

Результаты (2)



Исходное изображение



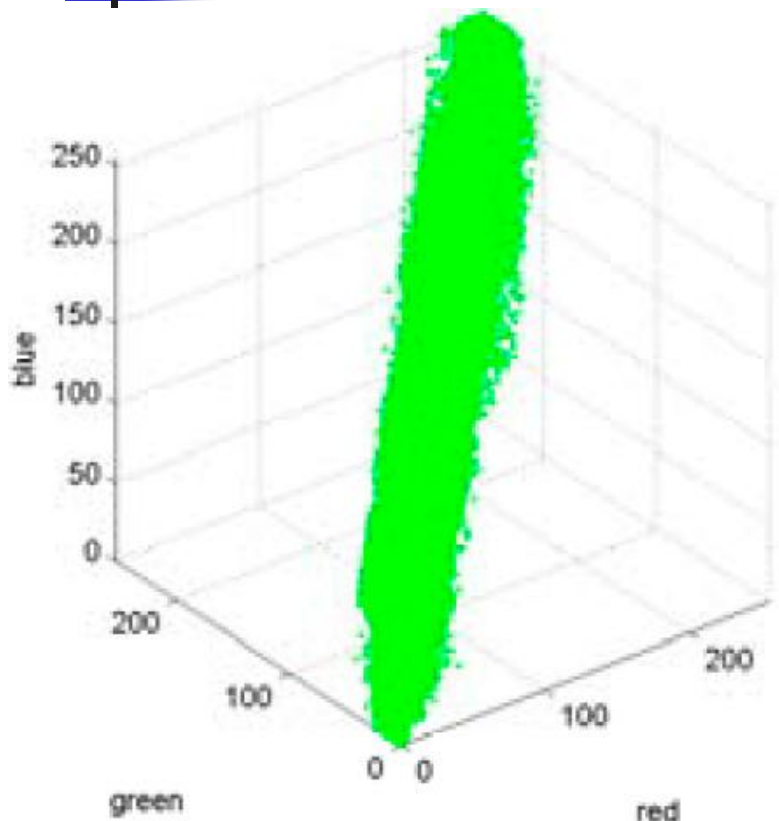
Целевое изображение



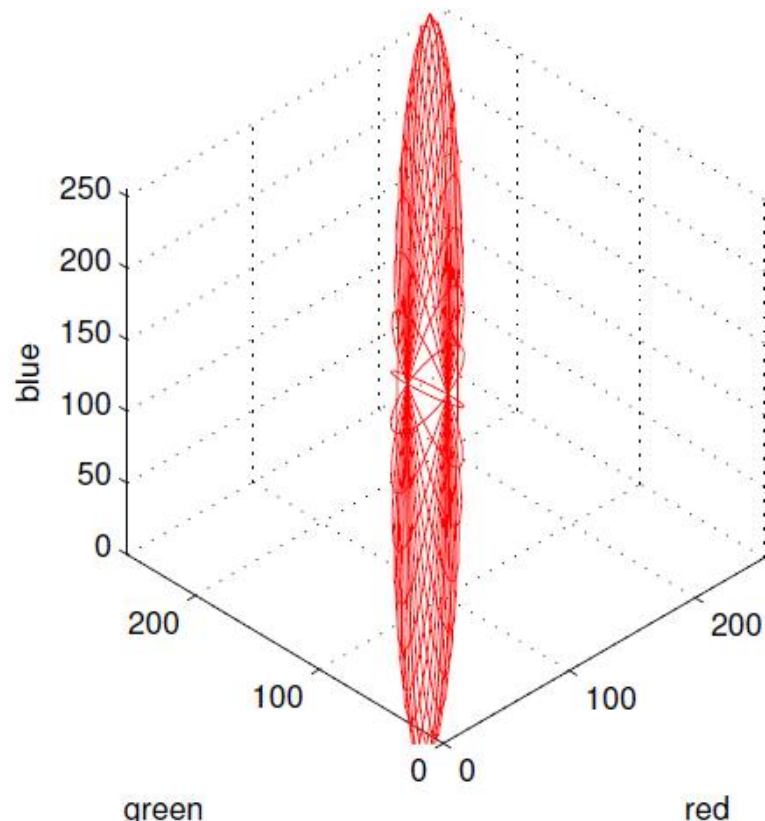
Результирующее
изображение

Global Color Transfer

Результаты (2)



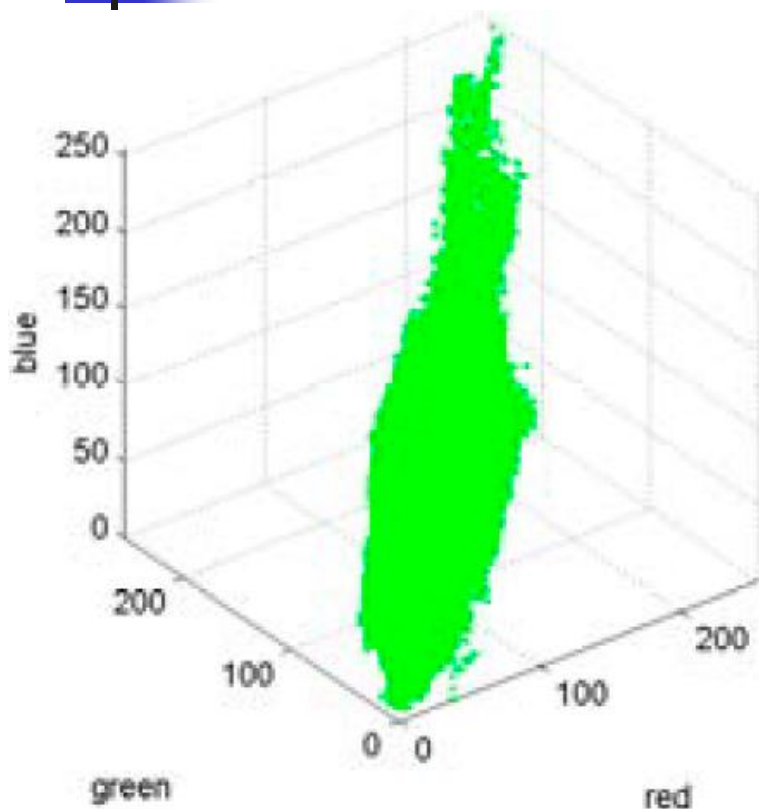
Распределение значений пикселей
исходного изображения в RGB-кубе



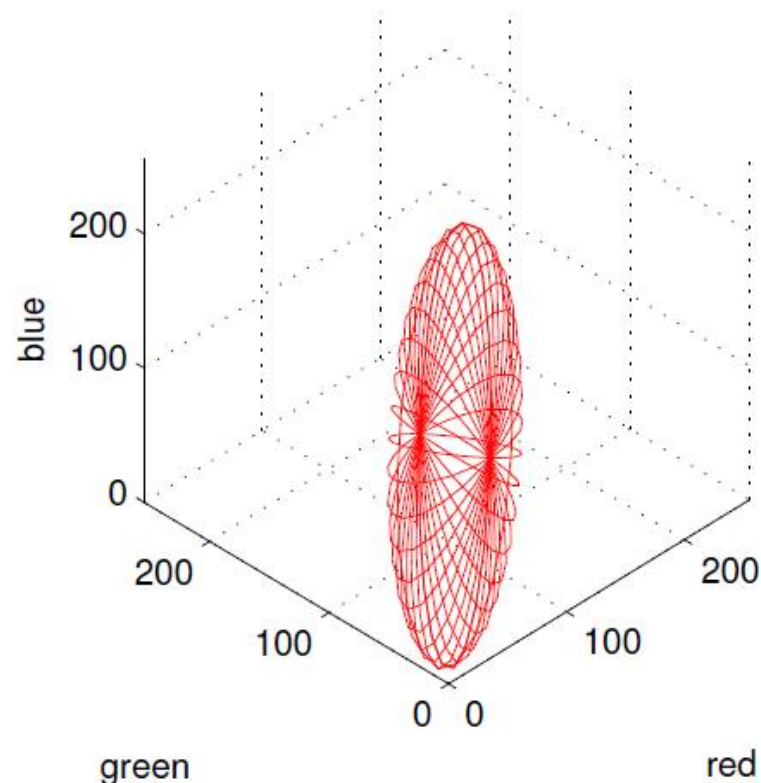
Эллипсоид, натянутый
на распределение значений пикселей

Global Color Transfer

Результаты (2)



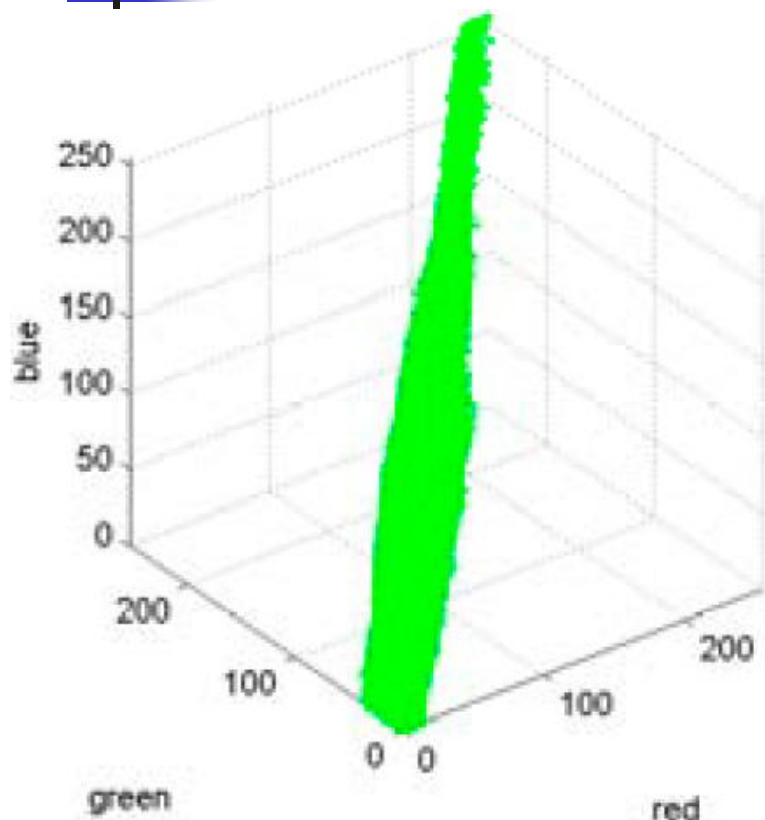
Распределение значений пикселей
целевого изображения в RGB-кубе



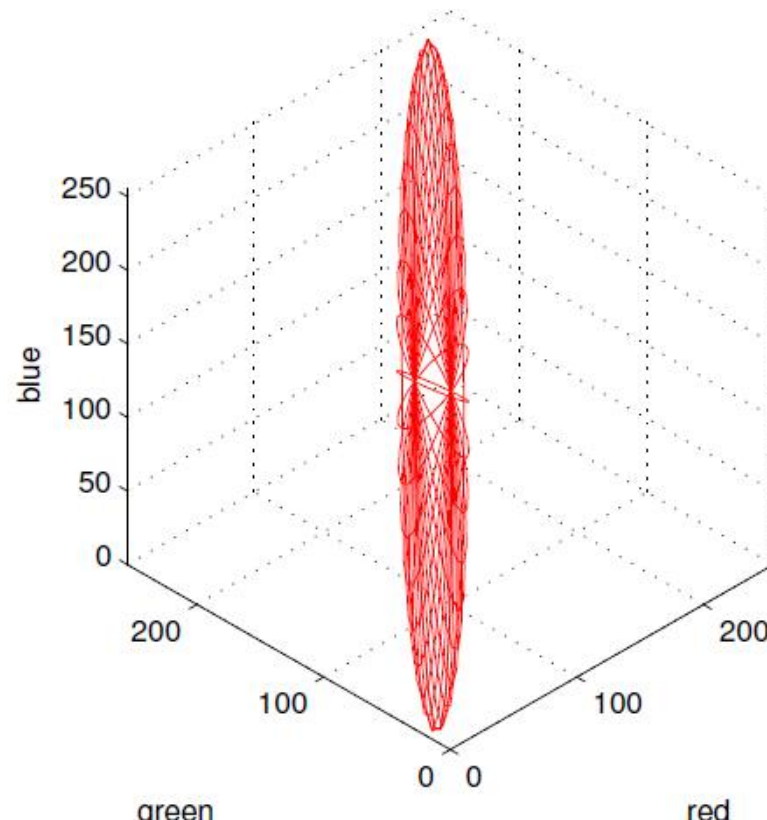
Эллипсоид, натянутый
на распределение значений пикселей

Global Color Transfer

Результаты (2)



Распределение значений пикселей
результатирующего изображения в RGB-кубе



Эллипсоид, натянутый
на распределение значений пикселей

Global Color Transfer

Выводы

Достоинства:

- Простота реализации
- Малая вычислительная сложность
- Используется совместное преобразование по трем каналам

Недостатки:

- Не учитываются локальные особенности изображения
- Недостаточность линейной модели преобразования



Содержание

- Введение
- Методы цветокоррекции
 - Глобальные
 - Histogram Matching
 - Global Color Transfer
 - **Кластерный метод**
 - Локальные
- Обнаружение бликов
- Заключение

Кластерный метод

- **Задача:**
Выполнить цветокоррекцию ракурсов, основываясь на выделении цветовых кластеров в ракурсах
- **Вход:**
Многоракурсное видео
- **Выход:**
Видео с исправленными по цвету ракурсами



Шаги алгоритма

1. Выбор ключевых кадров
2. Проведение вероятностной кластеризации пикселей
3. Вычисление параметров цветокоррекции
4. Интерполяция параметров цветокоррекции в неключевых кадрах
5. Выполнение цветокоррекции

Кластеризация

- Определяем N – количество цветовых кластеров
- Для каждого ракурса итеративно вычисляем вероятность P_{xy}^i принадлежности пикселя в позиции (x, y) к i -ому кластеру
- Введем обозначения:
 $I(x, y)$ – значение пикселя в пространстве CIE Lab
 $Dist$ – евклидово расстояние между значениями пикселей

Кластеризация

Вычисление вероятностей

На каждом шаге итерации для всех пикселей ракурса вычисляем вероятность:

$$iP_{xy} = \frac{\sum_{(x',y') \in N} iP_{x'y'} w((I(x,y), I(x',y')))}{\sum_{(x',y') \in N} w((I(x,y), I(x',y')))}$$

Функция расстояния между значениями пикселей:

$$w((I(x,y), I(x',y'))) = 1 - \frac{1}{1 + e^{-0.5(Dist(I(x,y), I(x',y')) - T_1)}}$$

N – пиксели, соседние с пикселем (x, y)

T_1 – параметр

Вычисление параметров

После того, как были посчитаны iP_{xy} для каждого ракурса, вычисляем параметры кластеров:

- Среднее значение пикселей для i -го кластера:

$$\mu_i = \sum_{x,y} iP_{xy} \cdot I(x, y) / \left(\sum_{x,y} iP_{xy} \right)$$

- Стандартное отклонение пикселей для i -го кластера:

$$\sigma_i = \sqrt{\sum_{x,y} iP_{xy} \cdot (I(x, y) - \mu_i)^2 / \left(\sum_{x,y} iP_{xy} \right)}$$

Вычисление параметров

Определение основных кластеров

- Для каждого ракурса вычисляем вероятность i -го кластера (его вес):

$$p_i = \sum_{x,y} i P_{xy} / \left(\sum_i \sum_{x,y} i P_{xy} \right)$$

- Сортируем p_i по убыванию и определяем M доминантных кластеров, максимизируя M , удовлетворяющее неравенству:

$$\left(\sum_{i=1}^M p_i \right) < T_2$$

T_2 – параметр

Цветокоррекция

Выполняем сведение цветов ракурсов к reference-ракурсу следующим образом:

$$I^{out}(x, y) = \sum_{i=1}^M p_i \left(\frac{\sigma_i^{ref}}{\sigma_i^{in}} (I^{in}(x, y) - \mu_i^{in}) + \mu_i^{ref} \right) / \left(\sum_{i=1}^M p_i \right)$$

μ_i^{in}, μ_i^{ref} – средние значения i -го кластера в текущем и reference-ракурсе

$\sigma_i^{in}, \sigma_i^{ref}$ – стандартные отклонения i -го кластера

$I^{in}(x, y), I^{out}(x, y)$ – исходные и преобразованные значения пикселей текущего ракурса

p_i – вероятность i -го кластера в текущем ракурсе

Кластерный метод

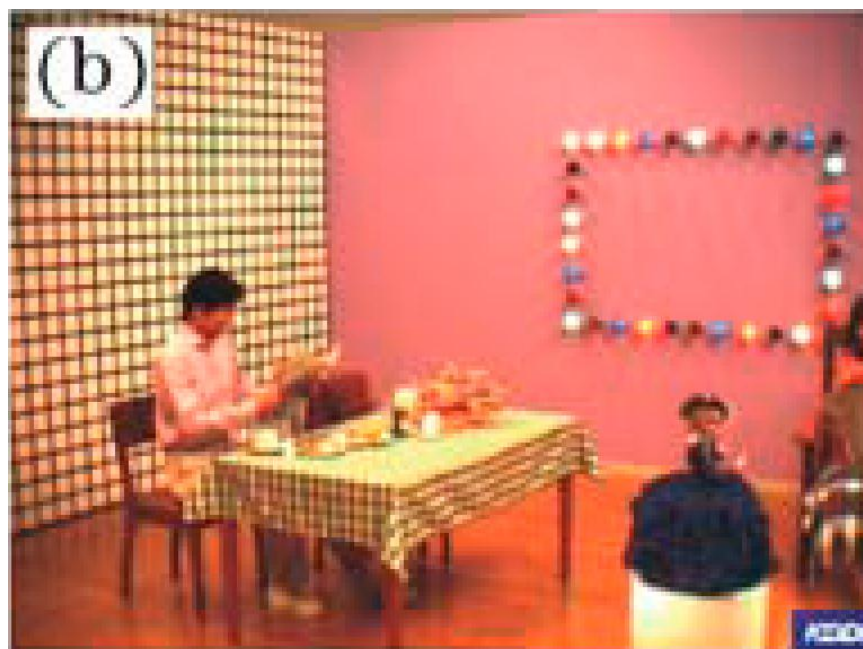
Результаты (1)



Левый ракурс

Кластерный метод

Результаты (1)



Правый ракурс

Кластерный метод

Результаты (1)



Правый скорректированный ракурс

Кластерный метод

Результаты (1)



Левый ракурс

Кластерный метод

Результаты (2)



Правый ракурс

Кластерный метод

Результаты (2)



Правый скорректированный ракурс

Кластерный метод

Результаты (2)



Левый ракурс

Кластерный метод

Выводы

Достоинство:

- Устойчивость во времени

Недостатки:

- Не учитывает положения пикселя в кадре
- Может неправильно корректировать маленькие объекты, сильно отличающиеся по цвету от всей сцены



Содержание

- Введение
- Методы цветокоррекции
 - Глобальные
 - Локальные
 - **Block-based method**
 - Local color correction for underwater stereo images
- Обнаружение бликов
- Заключение

Block-based method

- **Задача:**
Выполнить поблочную цветокоррекцию между центральным и боковыми ракурсами
- **Вход:**
Многоракурсное изображение
- **Выход:**
Исправленные по цвету ракурсы

Block-based method

Алгоритм

1. Оценка движения между центральным и боковыми ракурсами
2. Построение функции преобразования для каждого блока в каждом ракурсе для каналов Y , U , V
3. Применение полученных функций преобразования

Block-based method

Оценка движения

Выполняем оценку движения для блоков между центральными и боковыми ракурсами по каналу Y , в качестве метрики берем коэффициент корреляции:

$$NCC(i, j) = \frac{\sum_{x=x_0}^{x_0+N-1} \sum_{y=y_0}^{y_0+N-1} (Y_{anc}(x, y) - m_{anc})(Y_{view}(x-i, y-j) - m_{view})}{\sqrt{\sum_{x=x_0}^{x_0+N-1} \sum_{y=y_0}^{y_0+N-1} (Y_{anc}(x, y) - m_{anc})^2 \cdot \sum_{x=x_0}^{x_0+N-1} \sum_{y=y_0}^{y_0+N-1} (Y_{view}(x-i, y-j) - m_{view})^2}}$$

и максимизируем его в некоторой области $(i, j) \in A$

N – размер блока

m_{view} и m_{anc} – средние значения интенсивности в блоке

Y_{view} и Y_{anc} – значения интенсивности по каналу Y

Block-based method

Функция коррекции блоков

Для каждого блока по каждому каналу Y , U , V ищем функцию преобразования в виде полинома 3 степени от 3 переменных:

$$\begin{aligned} Y^{cor} = & a_{Y1}Y + a_{Y2}U + a_{Y3}V + a_{Y4}Y^2 + a_{Y5}U^2 + a_{Y6}V^2 \\ & + a_{Y7}YU + a_{Y8}YV + a_{Y9}UV + a_{Y10}Y^3 + a_{Y11}U^3 \\ & + a_{Y12}V^3 + a_{Y13}YUV + a_{Y14}Y^2U + a_{Y15}Y^2V + a_{Y16}YU^2 \\ & + a_{Y17}YV^2 + a_{Y18}U^2V + a_{Y19}UV^2 + a_{20} \end{aligned}$$

Block-based method

Функция коррекции блоков (2)

Запишем функцию преобразования
в матричном виде: $\mathbf{Y}^{\text{cor}} = \Psi \mathbf{a}_Y$

$$\Psi = [\Psi_1, \Psi_2, \dots, \Psi_{N^2}]^T$$

$$\Psi_i = \begin{bmatrix} Y_i & U_i & V_i & Y_i^2 & U_i^2 & V_i^2 & Y_i U_i & Y_i V_i & U_i V_i & Y_i^3 & U_i^3 & V_i^3 & \dots \\ Y_i U_i V_i & Y_i^2 U_i & Y_i^2 V_i & Y_i U_i^2 & Y_i V_i^2 & U_i^2 V_i & U_i V_i^2 & 1 \end{bmatrix}$$

Y_i, U_i, V_i – значения пикселей в блоке исходного ракурса

$$\mathbf{a}_Y = [a_{Y1}, a_{Y2} \dots a_{Y20}]$$

a_Y, a_U, a_V – векторы коэффициентов, определяющие функцию преобразования каждого канала

Block-based method

Функция коррекции блоков (3)

Коэффициенты $\mathbf{a}_Y = [a_{Y1}, a_{Y2} \dots a_{Y20}]$ находим, решая СЛАУ $\mathbf{Y}^{\text{avg}} = \Psi \mathbf{a}_Y + \boldsymbol{\varepsilon}$ методом наименьших квадратов, где

$$\mathbf{Y}^{\text{avg}} = \frac{1}{M} \sum_{i=1}^M \mathbf{Y}_i$$

\mathbf{Y}_i – значения интенсивностей пикселей в i -ом ракурсе
 M – количество ракурсов

$\boldsymbol{\varepsilon}$ – вектор ошибок

Block-based method

Результаты



Четвертый исходный ракурс

Colin Doutre, Panos Nasiopoulos, "Color Correction Preprocessing for Multiview Video Coding," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, 2009

Block-based method

Результаты



Третий исходный ракурс

Colin Doutre, Panos Nasiopoulos, "Color Correction Preprocessing for Multiview Video Coding," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, 2009

Block-based method

Результаты



Третий скорректированный ракурс

Colin Doutre, Panos Nasiopoulos, "Color Correction Preprocessing for Multiview Video Coding," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, 2009

Block-based method

Результаты



Четвертый скорректированный ракурс

Colin Doutre, Panos Nasiopoulos, "Color Correction Preprocessing for Multiview Video Coding," IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, 2009

Block-based method

Выводы

Достоинства:

- Низкая вычислительная сложность
- Преобразование зависит от положения блока в кадре

Недостатки:

- Неадекватность оценки движения для областей, различных по цвету и без текстуры
- Преобразования для блоков выполняются независимо



Содержание

- Введение
- Методы цветокоррекции
 - Глобальные
 - Локальные
 - Block-based method
 - **Local color correction for underwater stereo images**
- Обнаружение бликов
- Заключение

Постановка задачи

- **Задача:**

Выполнить цветокоррекцию подводного стереовидео, используя совместное преобразование по трем каналам RGB

- **Вход:**

- Стереопара
- Карта диспаратета

- **Выход:**

Скорректированные по цвету ракурсы



Алгоритм

Шаги алгоритма:

1. Выбор ракурса, который будет сводиться по цвету к другому ракурсу
2. Поиск и применение глобального цветового преобразования совместно по трем каналам к выбранному ракурсу
3. Выполнение локальной цветокоррекции



Выбор ракурса

- В качестве ракурса для преобразования (Source Image) выбираем тот, в котором количество различных цветов в RGB кубе больше
- Другой ракурс обозначаем как Reference Image

Глобальное преобразование

Составление векторов

Составляем векторы значений пикселей для

- Source Image: $X = (x_1, \dots, x_N)^T$
- Reference Image: $Y = (y_1, \dots, y_N)^T$

x_i , y_i – значения i -го пикселя в Source Image и значение соответствующего ему i -го пикселя в Reference Image в пространстве CIE Lab

Глобальное преобразование

Первый метод

- Ищем глобальное преобразование для X в виде $R \cdot x_i + t$
 R – матрица
 t – вектор сдвига
- Параметры преобразования получаем, используя итеративный алгоритм ближайших точек, минимизируя функцию

$$f_{ICP} = \sum_i \|R \cdot x_i + t - y_i\|^2$$

Итеративный алгоритм ближайших точек



Одна итерация:

1. Связывание точек по критерию ближайшего соседа (поиск для каждой точки множества X пары из множества Y)
2. Вычисление преобразования (смещение + поворот), минимизирующее среднеквадратичное расстояние между парными точками
3. Применение рассчитанного преобразования к множеству X и пересчет ошибки

Глобальное преобразование

Второй метод

Используя алгоритм Coherent Point Drift (CPD), ищем глобальное преобразование в более общем виде:

$$T(Y, \nu) = Y + \nu(Y)$$

ν – функция смещения, которая ищется с помощью минимизации выражения:

$$f_{\text{CPD}}(\nu, \sigma^2) = - \sum_{n=1}^N \log \sum_{m=1}^M \exp\left(\frac{\|x_n - y_m\|^2}{2\sigma^2}\right) + \frac{\lambda}{2} \varphi(\nu)$$

Глобальное преобразование

Второй метод (2)

$$f_{\text{CPD}}(\nu, \sigma^2) = - \sum_{n=1}^N \log \sum_{m=1}^M \exp\left(\frac{\|x_n - y_m\|^2}{2\sigma^2}\right) + \frac{\lambda}{2} \varphi(\nu)$$

- Отрицательная функция максимального

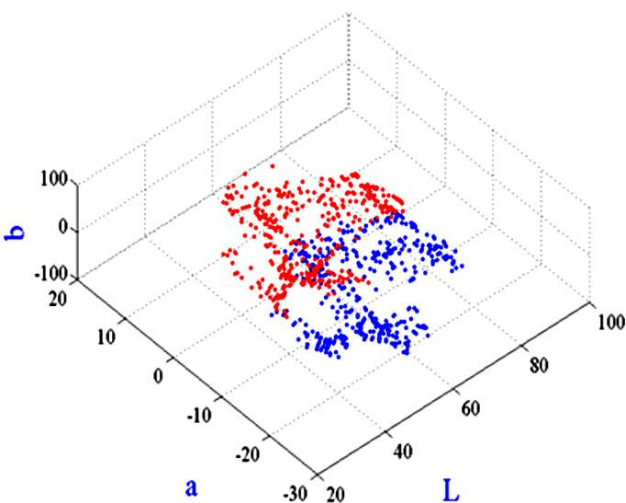
правдоподобия: $-\sum_{n=1}^N \log \sum_{m=1}^M \exp\left(\frac{\|x_n - y_m\|^2}{2\sigma^2}\right)$

- Регуляризационное правило: $\varphi(\nu)$

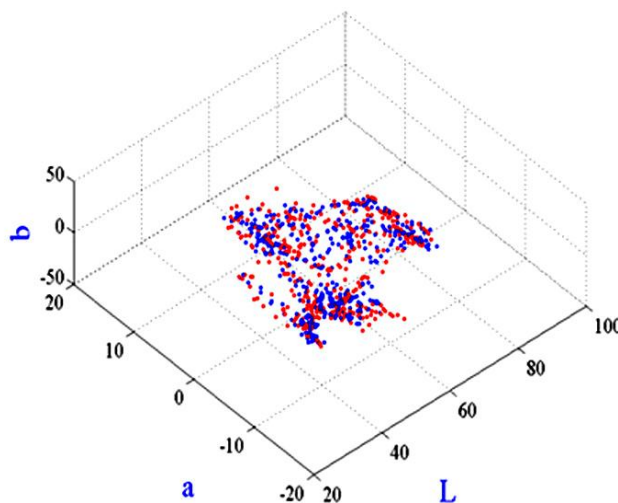
- Параметр: λ

- Ковариация, определяемая из expectation-maximization (EM) алгоритма: σ^2

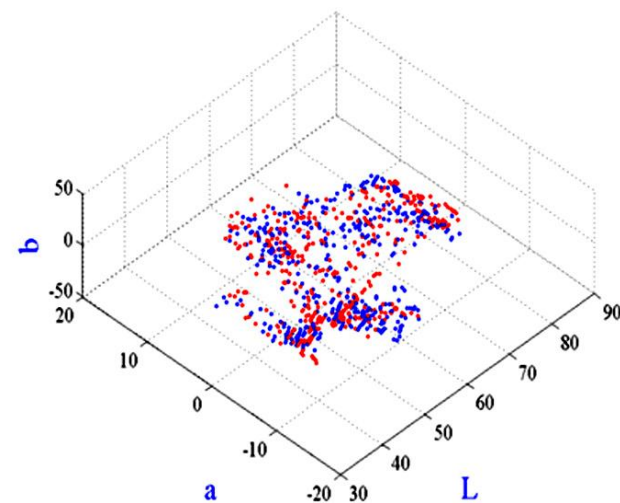
Сравнение двух методов Пространство CIE Lab



Распределение пикселей
до преобразования



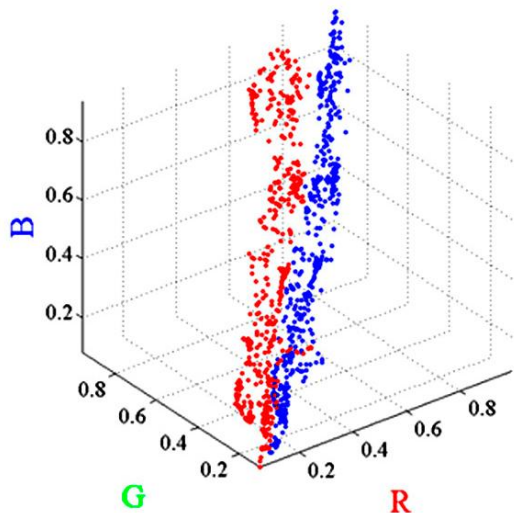
Распределение пикселей
после преобразования
(первый метод)



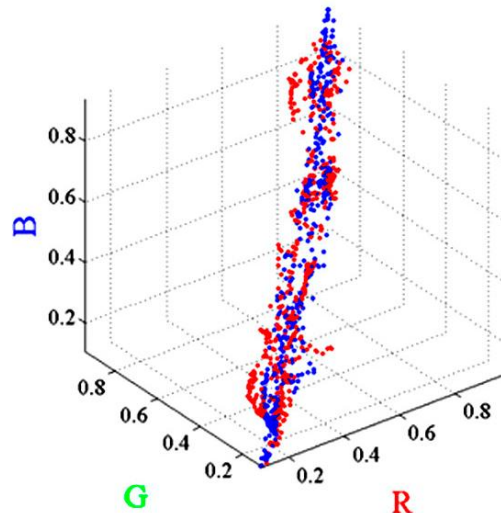
Распределение пикселей
после преобразования
(второй метод)

- Source Image
- Reference Image

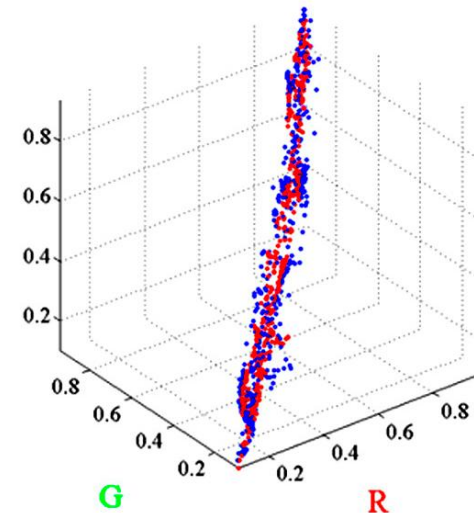
Сравнение двух методов Пространство RGB



Распределение пикселей
до преобразования



Распределение пикселей
после преобразования
(первый метод)



Распределение пикселей
после преобразования
(второй метод)

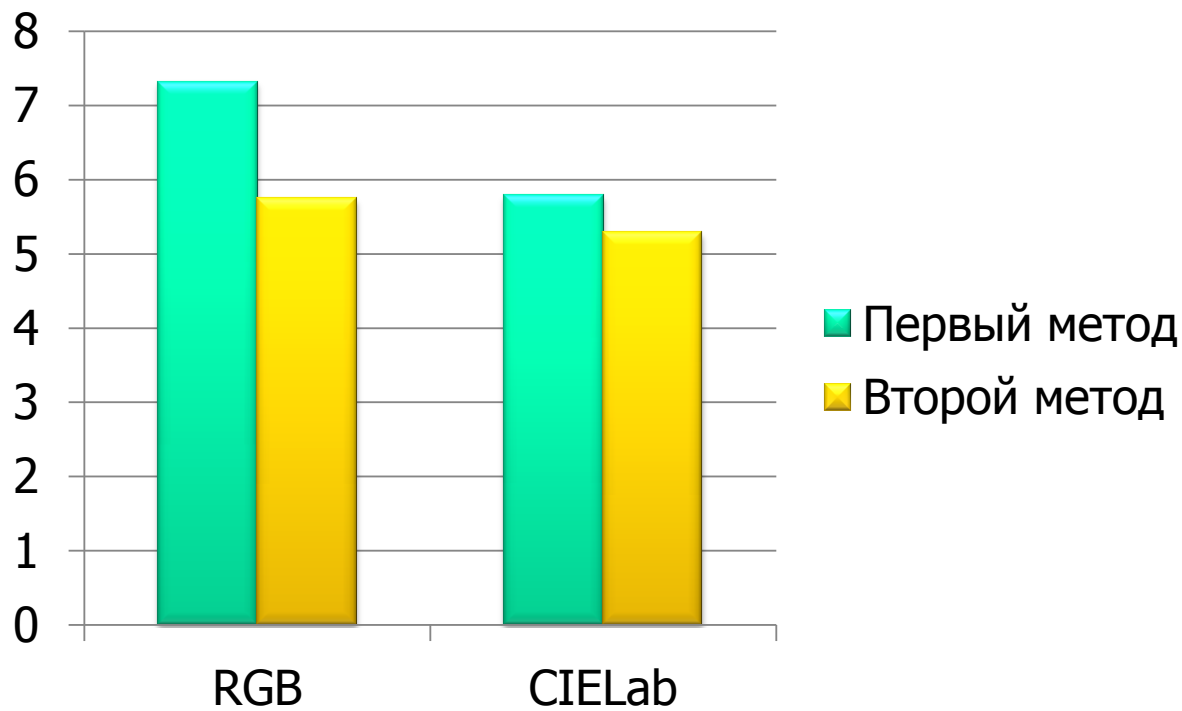
- Source Image
- Reference Image

Hyoungchul Shin et. al., "Local color correction with three dimensional point set registration for underwater stereo images," SPIE Optical Engineering, 2012

Сравнение двух методов

Численная оценка

Остаточные ошибки (residual errors) :



Локальная коррекция

Обозначения

$Y(p, q)$ – значение пикселя в Source Image на позиции (p, q) в пространстве CIE Lab

$X_Y(p, q)$ – значение пикселя в Reference Image, соответствующее пикселю $Y(p, q)$ в пространстве CIE Lab

$A(p, q) = X_Y(p, q) - Y(p, q)$ – вектор разницы значений соответствующих пикселей

Локальная коррекция

Преобразуем значение пикселя $Y(p, q)$:

$$\hat{Y}(p, q) = Y(p, q) + \hat{A}(i, j)$$

$$\hat{A}(p, q) = \frac{1}{N_T} \sum_{i, j \in W} D_c[Y(i, j)]$$

$$D_c[Y(i, j)] = \begin{cases} A(i, j), & \text{if } \|Y(i, j) - Y(p, q)\|^2 \leq T_d \\ 0, & \text{if } \|Y(i, j) - Y(p, q)\|^2 > T_d \end{cases}$$

W – локальная окрестность пикселя $Y(p, q)$,
состоящая из N_T пикселей

T_d – параметр алгоритма

Результаты

Правый ракурс



Hyoungchul Shin et. al., "Local color correction with three dimensional point set registration for underwater stereo images," SPIE Optical Engineering, 2012

Результаты

Левый ракурс



Hyoungchul Shin et. al., "Local color correction with three dimensional point set registration for underwater stereo images," SPIE Optical Engineering, 2012

Результаты

Скорректированный левый ракурс



Hyoungchul Shin et. al., "Local color correction with three dimensional point set registration for underwater stereo images," SPIE Optical Engineering, 2012

Результаты

Правый ракурс



Hyoungchul Shin et. al., "Local color correction with three dimensional point set registration for underwater stereo images," SPIE Optical Engineering, 2012



Выводы

Достоинства:

- Используется совместное преобразование по трем каналам
- Допускаются сильные искажения по цвету

Недостатки:

- Большая вычислительная сложность глобальной коррекции
- Локальная коррекция может вызвать размытие
- Предполагается геометрическая скалированность камер

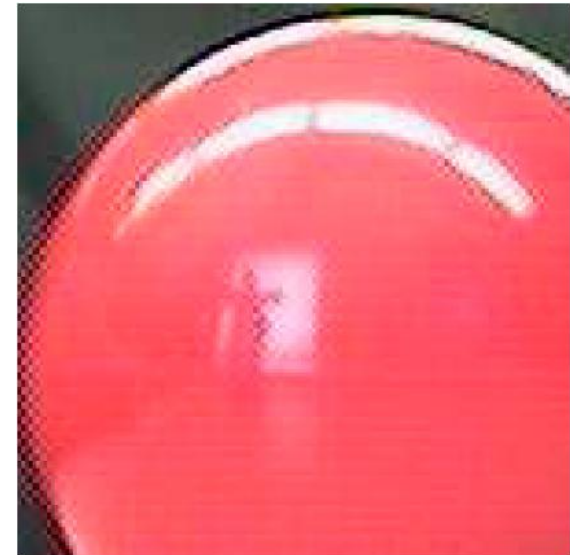


Содержание

- Введение
- Методы цветокоррекции
 - Глобальные
 - Локальные
- **Обнаружение бликов**
- Заключение

Классификация пикселей

- Матовые пиксели
- Отраженные ненасыщенные пиксели (значение интенсивностей которых не были «обрезаны» из-за конечной чувствительности камеры)
- Отраженные насыщенные



Обнаружение бликов

- **Задача:**

По двум изображениям одного объекта, снятого с одной точки, но при разных освещенностях, найти области содержащие отраженные (specular) пиксели

- **Вход:**

Reference Image с малым количеством отраженных пикселей и Target Image

- **Выход:**

Маска отраженных пикселей в Target Image

Обнаружение бликов

Примеры входных изображений



Reference Image



Target Image



Reference Image



Target Image

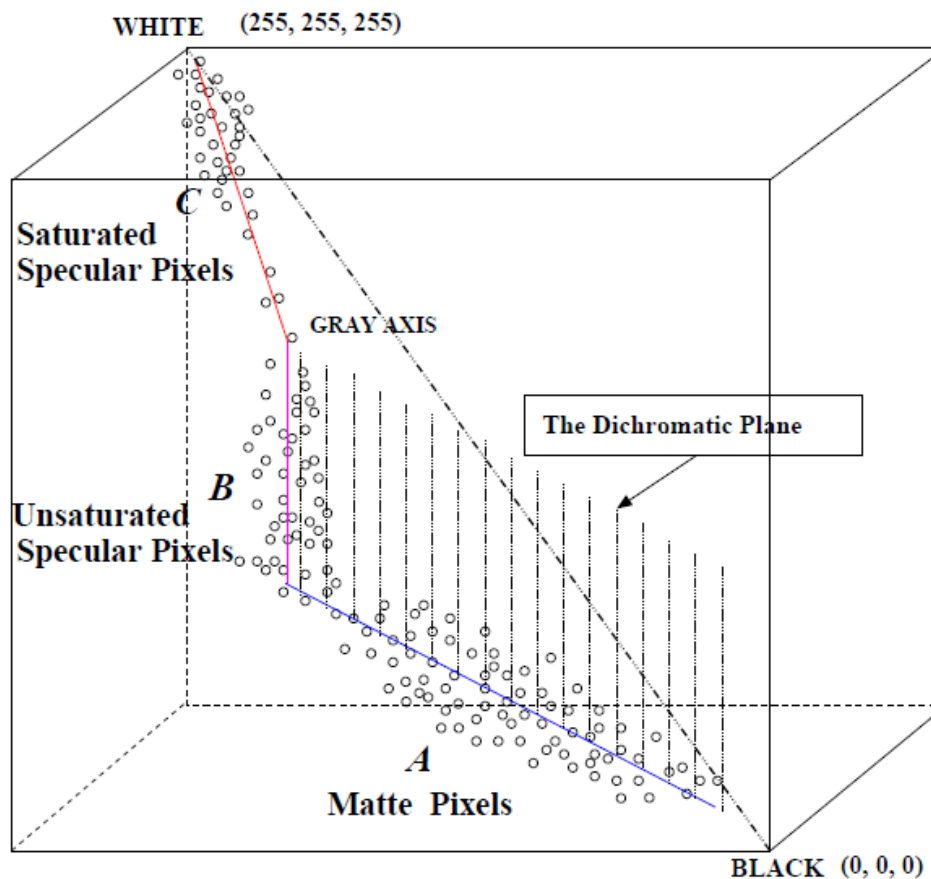
Jae Byung Park and Avinash C. Kak, "Truncated Least Squares Approach to the Detection of Specular Highlights in Color Images," IEEE Conference on Robotics and Automations, 2003

Шаги алгоритма

1. Находим порог в пространстве RGB, по которому определяем отраженные насыщенные пиксели в Target Image
2. Ищем линейную функцию преобразования, сводящую Reference Image к Target Image
3. Определяем отраженные ненасыщенные пиксели

Алгоритм

Поиск насыщенных пикселей



Пример распределения пикселей в RGB кубе

Jae Byung Park and Avinash C. Kak, "Truncated Least Squares Approach to the Detection of Specular Highlights in Color Images," IEEE Conference on Robotics and Automations, 2003

Алгоритм

Поиск насыщенных пикселей (2)

- Можно найти параллелограмм в пространстве RGB, возле которого сосредоточены все пиксели кроме насыщенных обрезанных
- Поэтому можно считать насыщенными пиксели, яркость которых выше порогового значения T
- Для поиска порогового значения T используем итеративный алгоритм, начиная с $T = 255$

Алгоритм

Поиск порогового значения

Шаг итерации:

1. Рассматриваем распределение пикселей ниже порогового значения
2. Для данного распределения методом главных компонент ищем величины собственных значений $\lambda_1 \geq \lambda_2 \geq \lambda_3$, определяющие размер аппроксимирующего эллипсоида
3. Уменьшаем пороговое значение до тех пор, пока λ_1 «резко не уменьшится»

Алгоритм

Поиск порогового значения (2)

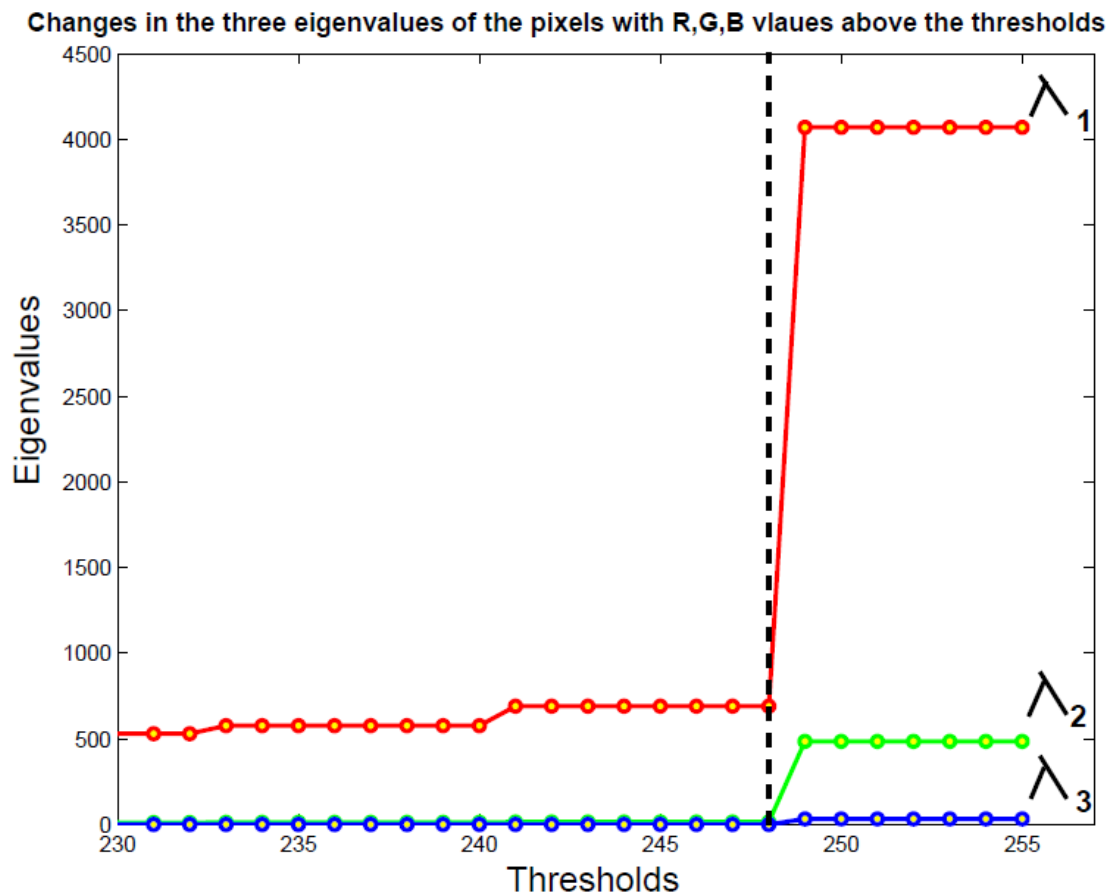


График зависимости собственных значений и порогового значения

Jae Byung Park and Avinash C. Kak, "Truncated Least Squares Approach to the Detection of Specular Highlights in Color Images," IEEE Conference on Robotics and Automations, 2003

Поиск линейного преобразования



- Пусть после отбрасывания насыщенных пикселей осталось k пикселей
- Определим матрицы:

$$\Omega = \begin{pmatrix} R_1 & R_2 & \cdots & R_k \\ G_1 & G_2 & \cdots & G_k \\ B_1 & B_2 & \cdots & B_k \end{pmatrix} \quad R_i, G_i, B_i - \text{значения } i\text{-го пикселя в Reference Image, в пространстве RGB}$$

Аналогично определим матрицу Ω' для Target Image

- Найдем методом наименьших квадратов матрицу M , минимизирующую соотношение $\|\Omega' - M \cdot \Omega\|^2$
- Матрица M определяет искомое преобразование

Jae Byung Park and Avinash C. Kak, "Truncated Least Squares Approach to the Detection of Specular Highlights in Color Images," IEEE Conference on Robotics and Automations, 2003

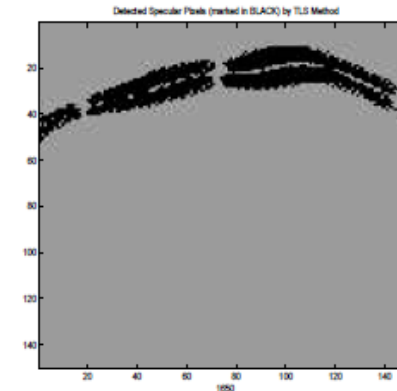
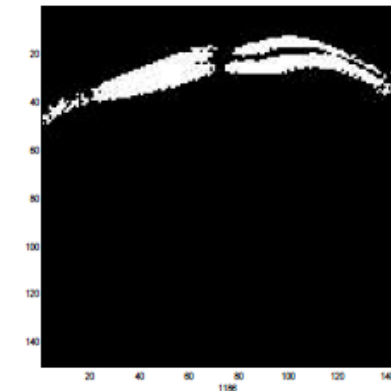
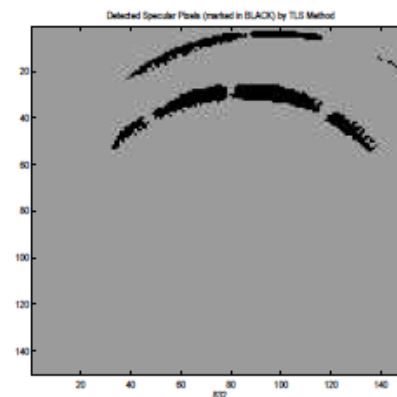
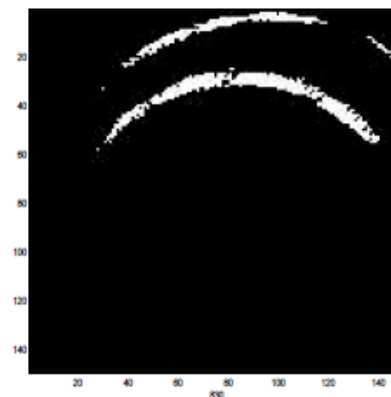
Поиск отраженных ненасыщенных пикселей



- Для Reference Image, используя оставшиеся пиксели, по каждому каналу R, G, B вычисляем вектор средних значений $\mu_k(\Omega)$ и вектор дисперсии $\sigma_k(\Omega)$, где k – канал R, G или B
- Для каждого оставшегося пикселя ω'_i в Target Image вычисляем обратное линейное преобразование $U = M^{-1} \cdot \omega'_i$
- Если $U_k \geq \mu_k(\Omega) + \alpha \sigma_k(\Omega)$ для какого-либо k , то i -ый пиксель считаем отраженным ненасыщенным, где α – параметр

Обнаружение бликов

Результаты



Reference
Image

Target Image

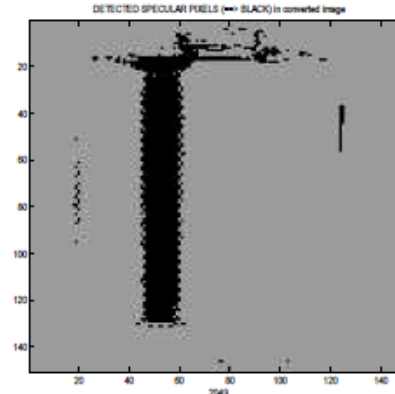
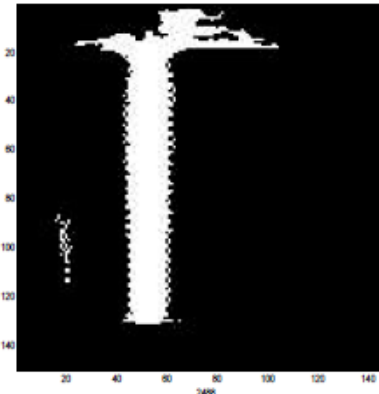
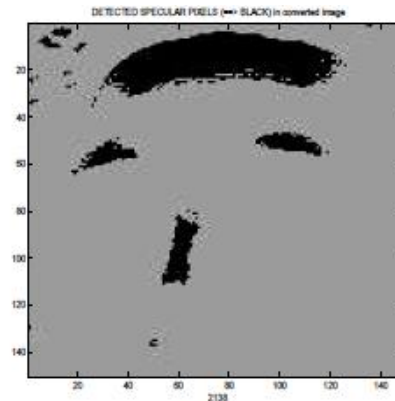
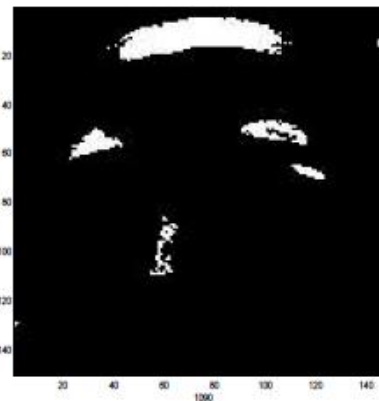
Субъективная
оценка

Результат

Jae Byung Park and Avinash C. Kak, "Truncated Least Squares Approach to the Detection of Specular Highlights in Color Images," IEEE Conference on Robotics and Automations, 2003

Обнаружение бликов

Результаты (2)



Reference
Image

Target Image

Субъективная
оценка

Результат

Jae Byung Park and Avinash C. Kak, "Truncated Least Squares Approach to the Detection of Specular Highlights in Color Images," IEEE Conference on Robotics and Automations, 2003

Обнаружение бликов

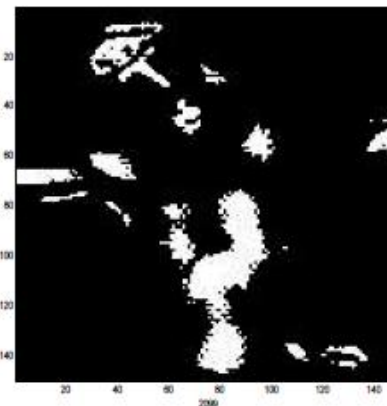
Результаты (3)



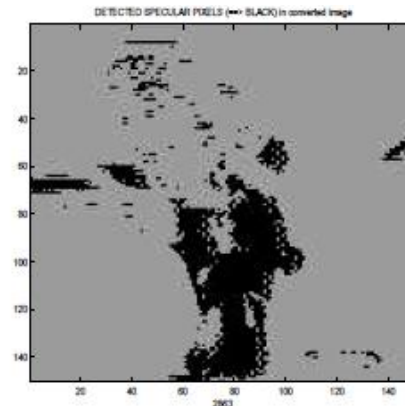
Reference
Image



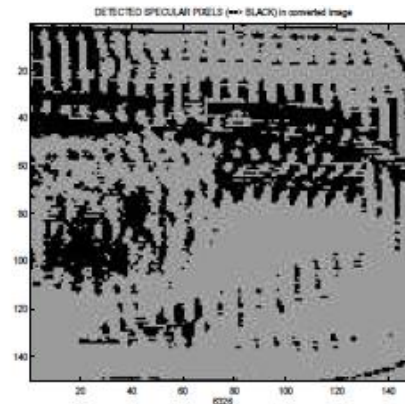
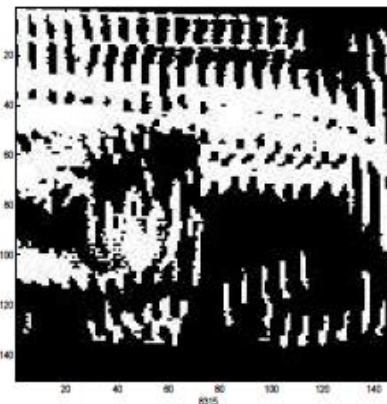
Target Image



Субъективная
оценка



Результат



Jae Byung Park and Avinash C. Kak, "Truncated Least Squares Approach to the Detection of Specular Highlights in Color Images," IEEE Conference on Robotics and Automations, 2003



Выводы

Достоинство:

- Работает на всех типах поверхностей

Недостаток:

- Требуется Reference Image без бликов



Содержание

- Введение
- Методы цветокоррекции
 - Глобальные
 - Локальные
- Обнаружение бликов
- **Заключение**

Заключение

Проделанная работа

- Реализована и улучшена авторская версия block-based алгоритма
- Реализован алгоритм исправления градиента

Block-based method

Правый ракурс



Block-based method

Левый ракурс



Block-based method

Левый скорректированный ракурс



Block-based method

Правый ракурс



Исправление градиента

Левый ракурс



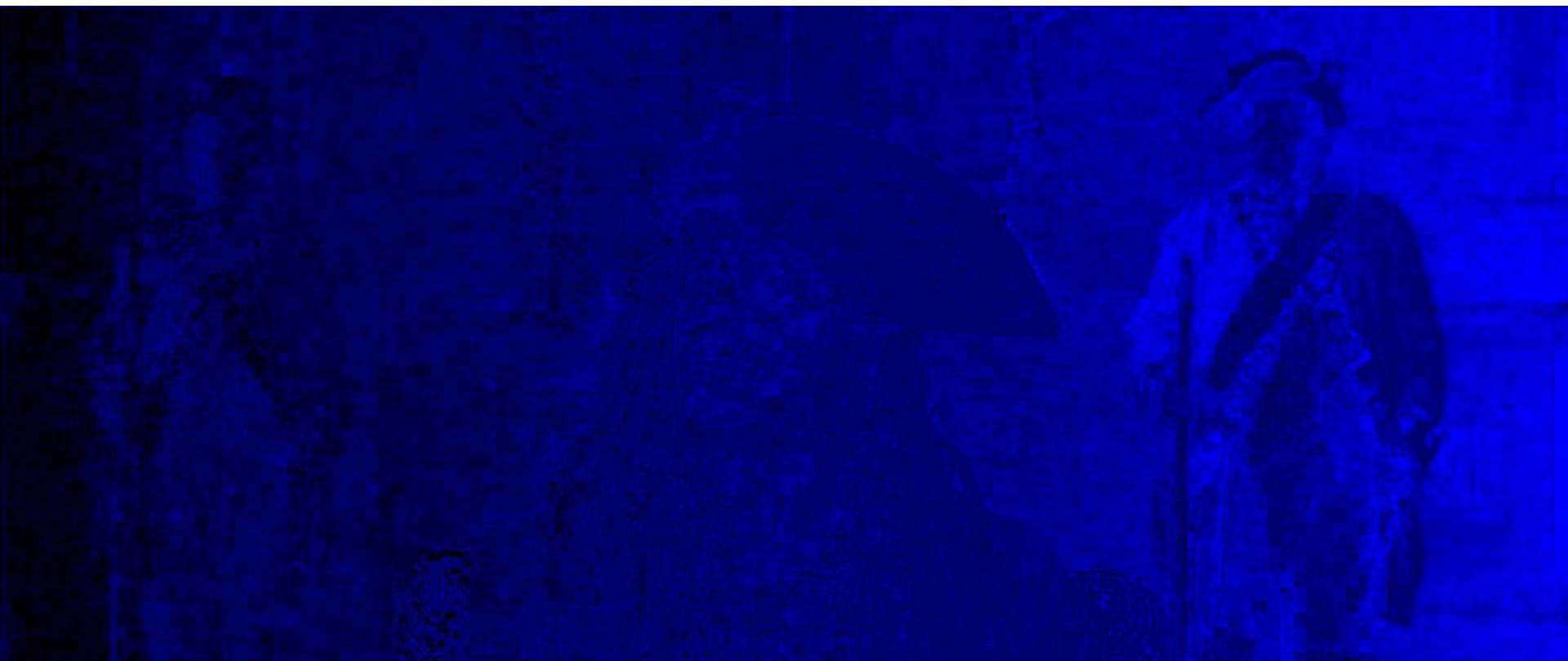
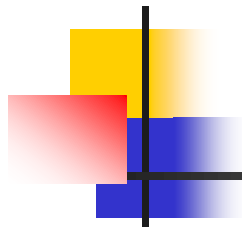
Исправление градиента

Правый ракурс



Исправление градиента

Скомпенсированная разница



Разница между левым и правым скомпенсированным ракурсом по синему каналу

Исправление градиента

График значений интенсивности

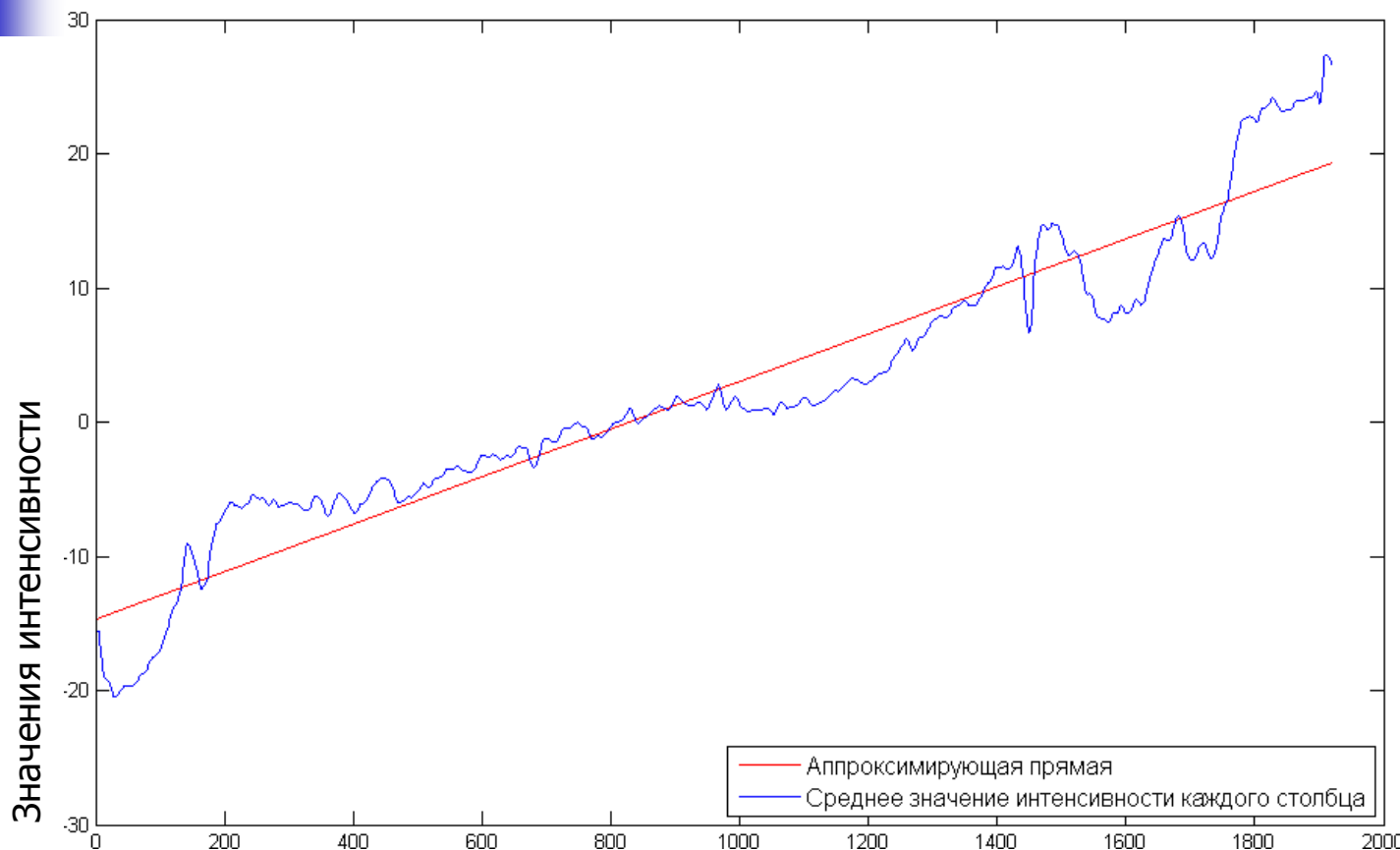


график средних значений интенсивности по столбцам для канала Blue
и их линейная аппроксимация для разницы между скомпенсированным левым
и правым ракурсом

Исправление градиента

Правый ракурс



Исправление градиента

Левый ракурс



Исправление градиента

Левый скорректированный ракурс



Исправление градиента

Правый ракурс



Исправление градиента (2)

Правый ракурс



Исправление градиента (2)

Левый ракурс



Исправление градиента (2)

Левый скорректированный ракурс



Исправление градиента (2)

Правый ракурс



Заключение

Дальнейшие планы

- Реализовать алгоритм построения 3D LUT для сведения ракурсов по цвету
- Провести анализ результатов, полученных при использовании 3D LUT
- Улучшение цветокоррекции:
 - Учет пространственного положения пикселей
 - Добавление алгоритма исправления градиентов
- Реализовать алгоритм обнаружения бликов

Литература

1. Xuezhong Xiao and Lizhuang Ma, "Color Transfer in Correlated Color - Space," *ACM International Conference on Virtual Reality Continuum and Its Applications*, Hong Kong, 2006, pp. 305-309.
2. Hyoungchul Shin, Ukil Yang and Kwanghoon Sohn "Local color correction with three dimensional point set registration for underwater stereo images," *Society of Photo-Optical Instrumentation Engineers*, April 2012.
3. Colin Doutre and Panos Nasiopoulos, "Color Correction Preprocessing for Multiview Video Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, September 2009, vol. 19, pp. 1400-1405.
4. Feng Shao, Mei Yu, Gang-yi Jiang and Ren-er Yang, "Color Correction for Multi-view Video Based on Color Variation Curve," *International Conference on Intelligent Computation Technology and Automation*, 2010, pp. 970-973.



Литература (2)

5. Jae Byung Park and Avinash C. Kak, "A Truncated Least Squares Approach to the Detection of Specular Highlights in Color Images," *IEEE International Conference on Robotics and Automations*, 2003.

Лаборатория компьютерной графики и мультимедиа



Видеогруппа — это:

- Выпускники в аспирантурах Англии, Франции, Швейцарии (в России в МГУ и ИПМ им. Келдыша)
- Выпускниками защищено 5 диссертаций
- Наиболее популярные в мире сравнения видеокодеков
- Более 3 миллионов скачанных фильтров обработки видео