**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра вычислительной техники**

**ОТЧЕТ**

**по лабораторной работе №4**

**по дисциплине «Методы обработки и анализа медицинских изображений»**

**Тема: ВИРТУАЛЬНАЯ ХРОМОЭНДОСКОПИЯ**

**Вариант 2**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 8308 |  | Сергеев В.С. |
|  |  | Петров Г.А. |
|  |  | Лукашов Н.С. |
| Преподаватель |  | Поздеев А.А. |

Санкт-Петербург

2023

**Цель работы**

Целью лабораторной работы является знакомство с современной технологией виртуальной хромоэндоскопии.

**Теоретические основания**

Одним из наиболее эффективных методов коррекции яркости и контраста является адаптивная эквализация гистограммы с ограничением по контрасту (Contrast Limited Adaptive Histogram Equalization – CLAHE) [1]. Эквализация гистограммы есть такое преобразование яркостей пикселов изображения, при котором гистограмма яркостей приводится в соответствие равномерному закону распределения.

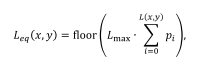
Адаптивная эквализация реализуется с помощью сканирующего окна и позволяет добиться повышения локального контраста. Для каждого анализируемого пиксела рассматривается только небольшая окрестность изображения, в которой осуществляется эквализация.

Для выполнения процедуры эквализации сперва рассчитывается нормализованная гистограмма распределения яркостей внутри окрестности:



где 𝐿max – в общем случае максимально возможная яркость (𝐿max = 1 в случае, когда яркость нормирована к диапазону [0…1]), 𝑛𝑖 – число пикселов с яркостью i, n – общее число пикселов, входящих в окрестность.

Новые значения яркости внутри окрестности определяются следующим образом:

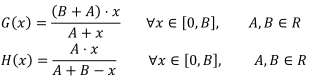


где floor() – функция округления вниз до целого, L(x, y) – яркость пиксела исходного изображения с координатами x, y внутри обрабатываемого блока.

В случае адаптивной эквализации в качестве результата преобразования берется значение центрального пиксела пересчитанной окрестности.

Для борьбы с шумами, уровень которых неизбежно повышается при контрастировании, в методе CLAHE применительно к каждой локальной области используется ограничение по контрасту. Суть этого этапа заключается в перераспределении значений яркости таким образом, чтобы число пикселов любой яркости не превышало величину заданного порога.

Одним из наиболее эффективных методов повышения контраста на эндоскопических изображениях является многомасштабное нелинейное контрастирование (Multi-Scale Image Contrast Enhancement – MSICE) [2]. Метод реализует повышение контраста, основываясь на следующих нелинейных функциях преобразования:



где х - входные данные, В - максимальное значение х (для изображений с глубиной цветового представления 8 бит на канал B = 255), А – коэффициент, который регулирует вид функции трансформации. Варьирование А позволяет получить различные нелинейные кривые и контролирует преобразование между входным значением х и выходным значением функций трансформации G(х) или Н(х) (Рис. 3). Как видно из рисунка 4 G(х) и Н(х) — это функции, реализующие противоположенное изменение входных данных. Функция G(х) увеличивает значение входных данных в соответствии с коэффициентом A, в то время как функция Н(х), наоборот, уменьшает значение входных данных.

Основная идея преобразования заключается в следующем: для каждого пикселя (i, j) исходного изображения разность между значением его яркости Yij и средней яркостью его окрестности Sij должна быть увеличена. Если верно условие Yij> Sij, то используют функцию G(х) для того, чтобы вычислить новое увеличенное значение яркости анализируемого пикселя и, таким образом, сделать разность между яркостью пикселя и средней яркостью его окрестности Sij больше. Если верно условие Yij < Sij, то используется функция Н(х) для того, чтобы вычислить новое уменьшенное значение яркости пикселя и, таким образом, увеличить разность между яркостью обрабатываемого пикселя и средней яркостью его окрестности Sij.

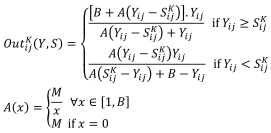
Среднее значение яркости окрестности пикселя SK для выбранного масштаба K рассчитывают следующим образом:



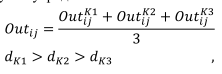
где dK - размер (в пикселях) окрестности для заданного масштаба К.

Значение параметра А определяет степень изменения исходного значения яркости пикселя Yij в зависимости от величины разности между Yij и SKij. Небольшое значение разности должно приводить к резкому изменению яркости обрабатываемого пикселя для увеличения локального контраста. Напротив, большие значения первоначальной разности между яркостью пикселя и средней яркостью окрестности должны обуславливать слабые изменения, так как контраст в этих случаях уже удовлетворительный.

Нелинейное изменение фактора А, а также сочетание уравнений (2.4), (2.5) и (2.6), приводят к следующей результирующей функции преобразования для нелинейного контрастирования.



где OutK – скорректированное значение яркости пикселя (i,j) для заданного масштаба К, М – константа, которая определяет степень контрастирования. Небольшие значения М приводят к выраженному контрастированию, большие – к умеренному. Полученное из проведенных авторами экспериментов, рекомендуемое значение для медицинских изображений М = 5000. Описанная процедура реализуется для окрестности трех разных размеров, результат получают путем усреднения:



где Outij значение яркости для пикселя (i, j), dK1, dK2 и dK3 размеры окрестности для трех разных пространственных масштабов, соответственно.

Три пространственных масштаба выбраны для обеспечения возможности влиять на нижние, средние и высокие частоты изображения. Проведенные экспериментальные исследования показали, что для изображения 1024 × 768 пикселей, dK1 должен быть примерно 110 пикселей по горизонтали. Выбор этой величины в качестве опорной точки, позволяет рассчитать размер области усреднения для других масштабов в соответствие с уравнениями:



где px и py – число столбцов и сток изображения.

**Задание**

В соответствии с индивидуальным заданием на моделирование необходимо разработать программу в виде консольного приложения, которая реализует обработку изображений, обеспечивающую эффект виртуальной хромоэндоскопии.

Студентам предлагается реализовать метод цифровой обработки, обеспечивающий эффект виртуальной хромоэндоскопии в соответствии с индивидуальным вариантом рисунок 1. Рекомендуется подобрать параметры используемых методов вручную, чтобы добиться лучшего визуального эффекта.

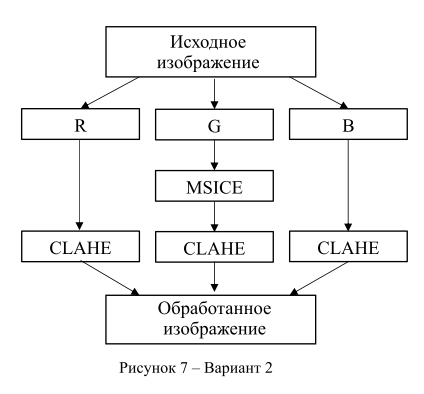


Рисунок 1 – Реализуемый алгоритм

**Ход работы**

Исходное цветовое изображение представлено на рисунке 2.

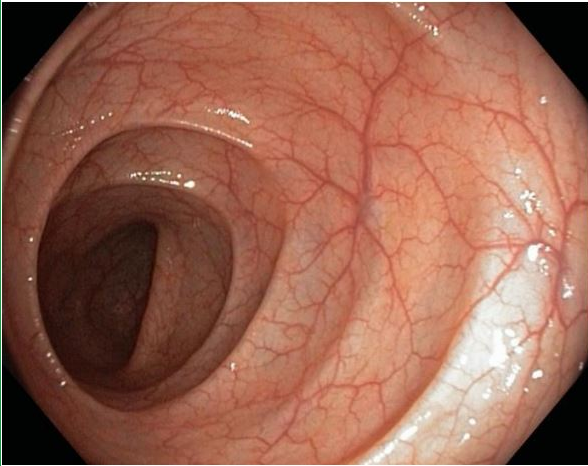


Рисунок 2 – Исходное цветовое изображение

Исходное изображение по каналам RGB представлено на рисунках 3-5.



Рисунок 3 – Исходное изображение канал R

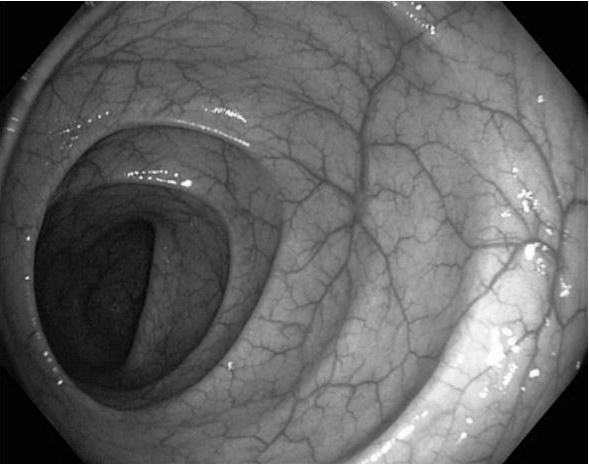


Рисунок 4 – Исходное изображение канал G

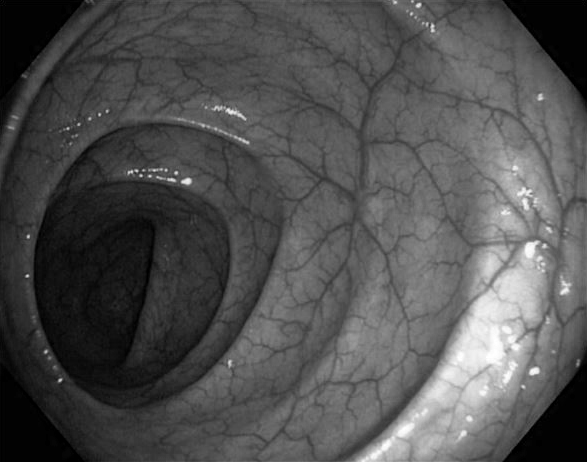


Рисунок 5 – Исходное изображение канал B

Канал G после применения к нему алгоритма MSICE представлен на рисунке 6. В качестве параметра M в алгоритме MSICE было выбрано значение – 5000.



Рисунок 6 – Канал G после MSICE

Результаты применения алгоритма CLAHE к каждому из каналов представлены на рисунках 7-9. В качестве ограничения контрастности в алгоритме CLAHE было выбрано значение – 3, а в качестве размера сетки – 4 на 4.

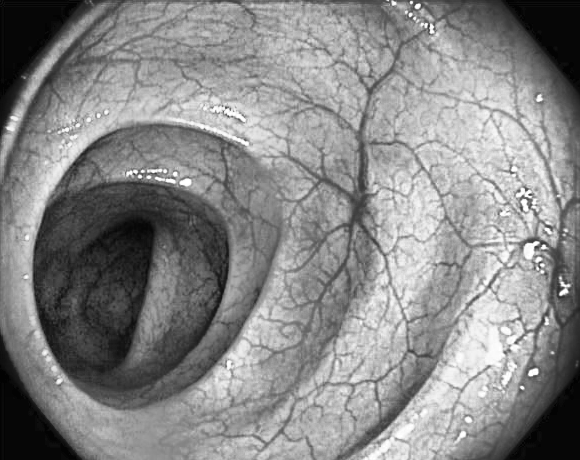


Рисунок 7 – Результат применения CLAHE к каналу R



Рисунок 8 – Результат применения CLAHE к каналу G



Рисунок 9 – Результат применения CLAHE к каналу B

Результирующее цветное изображение представлено на рисунке 10.

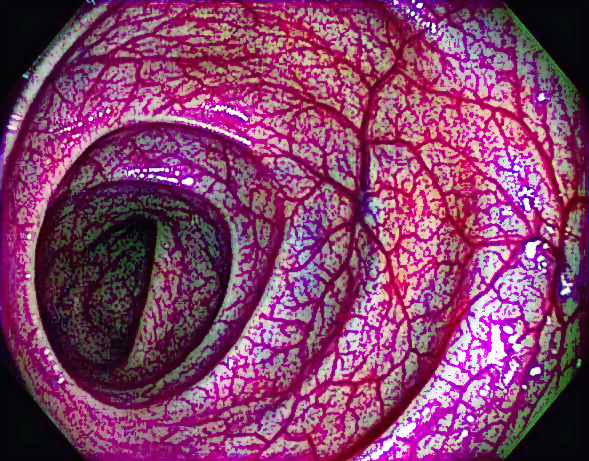


Рисунок 10 – Результирующее цветное изображение.

**Вывод**

В ходе выполнения лабораторной работы были изучены методы CLAHE и MSICE, а также реализованы на практике с визуализацией результатов на каждом шаге общего алгоритма. Так же было замечено, что для получения приемлемых результатов, необходимо тщательно подбирать параметры для каждого алгоритма.

Приложение А. Листинг исходного кода.

#include "opencv2/core/core.hpp"

#include "opencv2/features2d/features2d.hpp"

#include "opencv2/highgui/highgui.hpp"

#include "opencv2/calib3d/calib3d.hpp"

#include "opencv2/imgproc/imgproc.hpp"

#include <iostream>

**using** **namespace** cv;

**using** **namespace** std;

Mat MSICE(**const** Mat& Y, int M) {

Mat result = Mat::zeros(Y.size(), Y.type());

int dk[3];

dk[0] = Y.cols \* Y.rows / 7085;

dk[1] = dk[0] / 2;

dk[2] = dk[1] / 2;

Mat Ss[3];

**for** (int dk\_i = 0; dk\_i < 3; dk\_i++) {

blur(Y, Ss[dk\_i], Size(dk[dk\_i], dk[dk\_i]));

}

**for** (int i = 0; i < Y.rows; i++) {

**for** (int j = 0; j < Y.cols; j++) {

float out = 0;

uchar y = Y.at<uchar>(i, j);

**for** (int dk\_i = 0; dk\_i < 3; dk\_i++) {

uchar s = Ss[dk\_i].at<uchar>(i, j);

float A = (y - s) == 0 ? M : M / (y - s);

float out\_dki;

**if** (y >= s) {

out\_dki = (255 + A) \* y / (A + y);

}

**else** {

float A2 = (s - y) == 0 ? M : M / (s - y);

out\_dki = A \* y / (255 + A2 - y);

}

out += out\_dki;

}

out = out / 3;

result.at<uchar>(i, j) = out <= 0 ? 0 : (out >= 255 ? 255 : out);

*//result.at<uchar>(i, j) = out;*

}

}

normalize(result, result, 0, 255, NORM\_MINMAX);

**return** result;

}

int main()

{

int colorType = 1;

Mat img1 = imread("endo.jpg", colorType);

**if** (img1.empty()) {

**return** 0;

}

vector<Mat> rgb;

split(img1, rgb);

Mat gAfterMSICE;

vector<Mat> rgbAfterCLAHE = vector<Mat>(rgb.size());

gAfterMSICE = MSICE(rgb[1], 5000); *//менять параметр, 500, 5000(метода), 50000, 500000, 37500*

Ptr<CLAHE> clahe = createCLAHE(3, Size(4, 4)); *//2, Size(4, 4) 40, Size(8, 8)*

clahe->apply(rgb[2], rgbAfterCLAHE[2]);

clahe->apply(gAfterMSICE, rgbAfterCLAHE[1]);

clahe->apply(rgb[0], rgbAfterCLAHE[0]);

Mat final;

merge(rgbAfterCLAHE, final);

imshow("img1", img1);

imshow("R", rgb[2]);

imshow("G", rgb[1]);

imshow("B", rgb[0]);

imshow("G After MSICE", gAfterMSICE);

imshow("R After CLAHE", rgbAfterCLAHE[2]);

imshow("G After CLAHE", rgbAfterCLAHE[1]);

imshow("B After CLAHE", rgbAfterCLAHE[0]);

imshow("final", final);

waitKey();

**return** 0;

}