Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

Кирьяненко А.В. ADVANCED C++11/14 В приложениях дополненной реальности // Академия педагогических идей «Новация». Серия: Студенческий научный вестник. – 2018. – №7 (июль). – АРТ 439-эл. – 0,3 п.л. - URL: http: //akademnova.ru/page/875550

РУБРИКА: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ

УДК 004.588

Кирьяненко Александр Владиславович
Бакалавр
Московский государственный технический университет имени Н.Э. Баумана
г. Москва, Российская Федерация
e-mail: kiryanenkoav@mail.ru

ADVANCED C++11/14 В ПРИЛОЖЕНИЯХ ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Аннотация: В статье рассмотрены особенности использования C++11/14 в приложениях дополненной реальности. Также описывается работа с библиотекой алгоритмов компьютерного зрения и обработки изображений «OpenCV».

Ключевые слова: дополненная реальность, C++11/14, OpenCV, компьютерное зрение.

Kiryanenko Alexander Vladislavovich
Bachelor's degree
Bauman Moscow state technical university
Moscow, Russian Federation
e-mail: kiryanenkoav@mail.ru

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

ADVANCED C++11/14 IN APPLICATIONS OF ADDITIONAL REALITY

Abstract: The article discusses the features of using C ++ 11/14 in applications of additional reality. Also, work with the library of computer vision and image processing algorithms "OpenCV" is described.

Keywords: additional reality, C ++ 11/14, OpenCV, computer vision.

Дополненная реальность – работа с видео

OpenCV – (Open Computer Vision) — библиотека компьютерного зрения с открытым исходным кодом, предоставляющая набор типов данных и численных алгоритмов для обработки изображений алгоритмами компьютерного зрения [1]. Реализована на C/C++.

В библиотеке OpenCV имеются алгоритмы выделения линий из изображения:

1. **Преобразование Хаффа** – это линейное преобразование для обнаружения прямых (есть для элипсов и окружностей). Прямая может быть задана уравнением $y = mx + b$ и может быть вычислена по паре точек (x, y) на изображении. Основная идея преобразования Хафа — учесть характеристики прямой не как уравнение, построенное по паре точек изображения, а по её параметрам, то есть m — коэффициента наклона и b — точки пересечения с осью ординат. Таким образом, прямая, заданная уравнением $y = mx + b$, может быть представлена в виде точки с координатами (b, m) в пространстве параметров [2].
2. **LSD** (Line Segment Detector) – это детектор отрезков прямых с линейным временем поиска. Он быстрее и лучше ищет, чем преобразование Хаффа.

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

Для выделения фильтров, к изображению необходимо применить фильтры:

Фильтр Собеля — дискретный дифференциальный оператор, вычисляющий приближённое значение градиента яркости изображения. Результатом применения оператора Собеля в каждой точке изображения является либо вектор градиента яркости в этой точке, либо его норма [3].

```
void cvSobel(const CvArr* src, CvArr* dst, int xorder, int yorder, int aperture_size =
3)
```

Фильтр Канни – оператор обнаружения границ изображения. Использует многоступенчатый алгоритм для обнаружения широкого спектра границ в изображениях.

```
void Canny(InputArray image, OutputArray edges, double threshold1, double threshold2,
int apertureSize = 3, bool L2gradient = false)
```

Ниже представлен пример выделения линий используя LSD

Листинг 1 – пример выделения линий при помощи LSD

```
Ptr<LineSegmentDetector> ls = createLineSegmentDetector(LSD_REFINE_STD);
while (1) {
	Mat frame;
	std::vector<Vec4f> lines_std;
	bool bSuccess = cap.read(frame); // read a new frame from video
	if (!bSuccess) {
		std::cout << "Cannot read the frame from video file" << std::endl;
		break;
	}
	Mat gray_image;
	cvtColor(frame, gray_image, COLOR_BGR2GRAY);
	ls->detect(gray_image, lines_std); // Detect the lines
	ls->drawSegments(frame, lines_std);
	cv::resize(frame, frame, cv::Size(800, 600));
	imshow("MyVideo", frame); //show the frame in "MyVideo" window
	if (waitKey(30) == 27) {
		break;
	}
}
```

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

Рисунок 1 – Исходное изображение

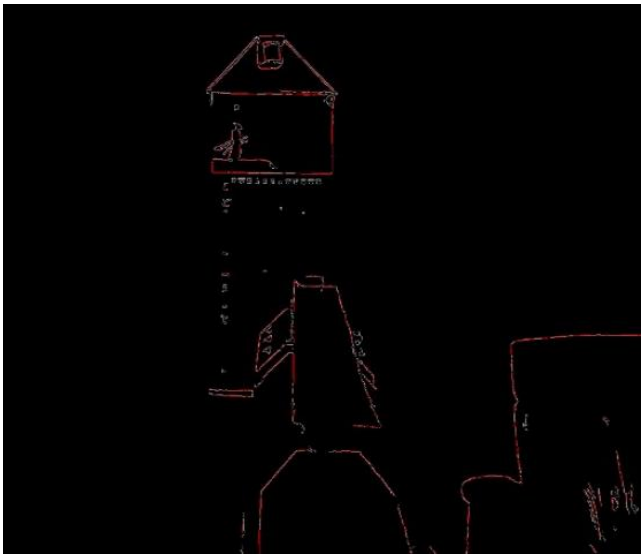

Рисунок 2 – Результат выделения линий используя LSD

Пример 1: необходимо реализовать программу “кротовая нора”, которая:

- читает видео файл “корридор”;
- выделить линии принцельной пола и потолка на изображении
- отрисовать линии пола и потолка поверх изображение зеленым цветом

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

Листинг 2 – программа “кротовая нора”

```
#include <iostream>
#include <opencv2/opencv.hpp>
#include <opencv2/highgui/highgui.hpp>
using namespace std;
using namespace cv;

int main(int argc, char* argv[]) {
	cout << "start" << endl;
	VideoCapture cap("video.mov"); // open the video file for reading
	if (!cap.isOpened()) return -1;
	double fps = cap.get(CV_CAP_PROP_FPS); //get the frames per seconds of the video
	cout << "Frame per seconds : " << fps << endl;
	namedWindow("MyVideo", CV_WINDOW_AUTOSIZE); //create a window called "MyVideo"

	Ptr<LineSegmentDetector> ls = createLineSegmentDetector(LSD_REFINE_STD);
	Mat edges;
	while (1) {
		Mat frame;
		bool bSuccess = cap.read(frame); // read a new frame from video
		if (!bSuccess) {
			cout << "Cannot read the frame from video file" << endl;
			break;
		}

		cvtColor(frame, edges, COLOR_BGR2GRAY); // Перевод в градации серого
		std::vector<Vec4f> lines_std;
		ls->detect(edges, lines_std); // Detect the line

		for (size_t i = 0; i < lines_std.size(); ++i) {
			Point2f p1, p2;
			p1.x = lines_std[i][0]; p1.y = lines_std[i][1];
			p2.x = lines_std[i][2]; p2.y = lines_std[i][3];
			line(frame, p1, p2, Scalar(25, 255, 25));
		}

		imshow("MyVideo", frame); //show the frame in "MyVideo" window
		if (waitKey(30) == 27) {
			break;
		}
	}
}
```

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

Рисунок 3 – Результат работы программы “кротовая нора”

Гистограммы

Для поиска похожих изображений, достаточно сравнить гистограммы этих изображений. Этот способ простой и малозатратный, однако обладает низкой точностью.

Для полутонового изображения гистограмма обладает следующими свойствами:

- это массив чисел
- каждый элемент массива содержит число точек с цветом, который попадает в соответствующий диапазон

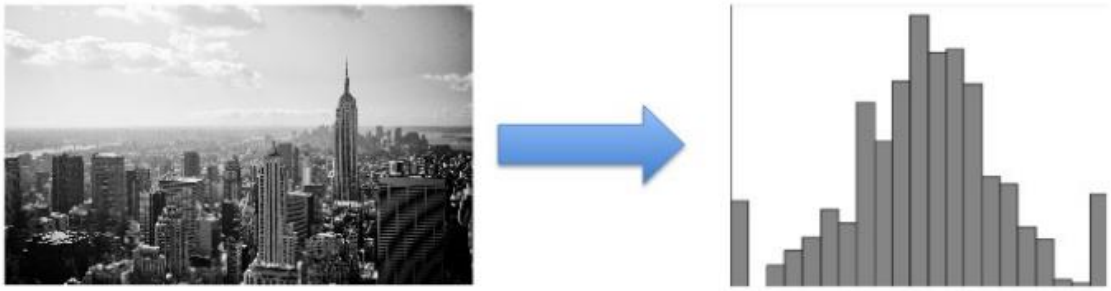

Рисунок 4 – Гистограмма для полутонового изображения

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

Пример 2 нужно написать программу для расчета гистограмм для набора изображений, которые находятся в папке. И написать функктор, позволяющий упорядочить набор изображений по степени похожести (сравниваем на основе гистограмм) на образец при помощи функции std::sort.

Листинг 3 – Программа определяющая схожесть изображений на основе гистограмм

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <opencv2/opencv.hpp>
using namespace std;
using namespace cv;

#define File_Name_Len 80

class Image {
	MatND m_hist;
	char m_name[File_Name_Len];

public:
	Image(const char *filename);
	Image(const Image &obj);
	const char *getName() const;
	MatND getHist() const;
};

Image::Image(const char *filename) {
	strcpy(m_name, filename);
	int histSize = 256; // bin size
	float range[] = { 0, 255 };
	const float *ranges[] = { range };
	Mat img = imread(filename, 0);
	// Calculate histogram
	calcHist(&img, 1, 0, Mat(), m_hist, 1, &histSize, ranges, true, false); }

const char *Image::getName() const {
	return m_name;
}

MatND Image::getHist() const {
	return m_hist;
}

Image::Image(const Image &obj) {
	strcpy(m_name, obj.getName());
	m_hist = obj.getHist();
}

// Функтор сортировки
```

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

```
class Similar {
	MatND m_cmpHist;

public:
	Similar(const char *cmpImage);
	bool operator() (Image &a, Image &b);
};

Similar::Similar(const char *cmpImage) {
	Mat grey = imread(cmpImage, 0);
	int histSize = 256; // bin size
	float range[] = { 0, 255 };
	const float *ranges[] = { range };
	// Calculate histogram
	calcHist(&grey, 1, 0, Mat(), m_cmpHist, 1, &histSize, ranges, true, false); }

bool Similar::operator()(Image &a, Image &b) {
	double cmpA = compareHist(a.getHist(), m_cmpHist, CV_COMP_CORREL);
	double cmpB = compareHist(b.getHist(), m_cmpHist, CV_COMP_CORREL);
	return cmpA > cmpB;
}

int main() {
	char imagesFileNames[][File_Name_Len] = {
		"images/box.JPG",
		"images/chair.JPG",
		"images/computer1.JPG",
		"images/computer2.JPG",
		"images/fire_extinguisher.JPG",
		"images/hanger.JPG",
		"images/keyboard.JPG",
		"images/monitor.JPG",
		"images/switches.JPG"
	};
	size_t count = sizeof imagesFileNames / File_Name_Len;

	vector<Image> images;
	for (size_t i = 0; i < count; ++i) {
		images.push_back(Image(imagesFileNames[i]));
	}

	Similar similaritySort("cmpImage.JPG");
	sort(images.begin(), images.end(), similaritySort);

	for (size_t i = 0; i < count; ++i) {
		cout << images[i].getName() << endl;
	}

	return 0;
}
```

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

Выделение контрастных объектов при помощи ступенчатого преобразования

Mat – ключевая структура в OpenCV представляет матрицу или изображение.

Пример 3: нужно реализовать программу “кротовая нора”, которая:

- читает изображение с доской и стикером;
- читает изображение “динозавра”
- находит розовый стикер на доске
- отображает динозавра “привязанного” к точке розового стикера

Как это реализовать:

1. Захватываем картинку из видеопотока
2. Выделяем объект реального мира на картинке
3. Рисуем на результирующем кадре наше изображение так, чтобы оно корректно располагалось относительно найденного объекта
4. Отображаем на экран результирующий кадр

Листинг 4 – Программа, отображающая динозавра между стикерами

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <opencv2/opencv.hpp>
#include <math.h>


// Функция записывает в stickersCoords координаты стикеров
void recogniseStickersByThreshold(cv::Mat image, std::vector<std::vector<cv::Point>>
&stickersCoords) {
	cv::Mat image_hsv;
	cv::cvtColor(image, image_hsv, cv::COLOR_BGR2HSV); // Преобразуем в hsv
	cv::Mat tmp_img(image.size(), CV_8U);
	// Выделение подходящих по цвету областей. Цвет задается константой :)
	cv::inRange(image_hsv, cv::Scalar(100, 100, 100), cv::Scalar(255, 255, 255),
                                                                    tmp_img);
	// "Замазать" огрехи в при выделении по цвету
	cv::dilate(tmp_img, tmp_img, cv::Mat(), cv::Point(-1, -1), 3);
	cv::erode(tmp_img, tmp_img, cv::Mat(), cv::Point(-1, -1), 1);
	//Выделение непрерывных областей
```

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

```
	cv::findContours(tmp_img, stickersCoords, CV_RETR_EXTERNAL,
CV_CHAIN_APPROX_NONE);
}

bool cmpPoint(const cv::Point &a, const cv::Point &b) {
	return a.x < b.x || a.x == b.x && a.y < b.y;
}

int main() {
	using namespace cv;
	using namespace std;

	cout << "start" << endl;
	VideoCapture cap("video.mov"); // open the video file for reading
	if (!cap.isOpened()) return -1;
	double fps = cap.get(CV_CAP_PROP_FPS); //get the frames per seconds of the video
	cout << "Frame per seconds : " << fps << endl;
	namedWindow("MyVideo", CV_WINDOW_AUTOSIZE); //create a window called "MyVideo"
	// Считываю изображение динозавра
	Mat dino = imread("dino.png");

	std::vector<std::vector<cv::Point>> stickersCoords;
	while (1) {
		Mat frame;
		bool bSuccess = cap.read(frame); // read a new frame from video
		if (!bSuccess) {
			cout << "Cannot read the frame from video file" << endl;
			break;
		}

		recogniseStickersByThreshold(frame, stickersCoords);
		cv::Point sticker1 = *max_element(stickersCoords.front().begin(),
						stickersCoords.front().end(), cmpPoint);
		cv::Point sticker2 = *min_element(stickersCoords.back().begin(),
						stickersCoords.back().end(), cmpPoint);


		// Масштабирование
		int side = abs(sticker1.x - sticker2.x);
		Mat imgForPaste;
		cv::resize(dino, imgForPaste, cv::Size(side, side));
		// Слияние коартинок
		Mat roi = frame(Rect(min(sticker1.x, sticker2.x), min(sticker1.y,
sticker2.y),
						side, side));	// подматрица frame
		imgForPaste.copyTo(roi);

		imshow("MyVideo", frame); //show the frame in "MyVideo" window
		if (waitKey(30) == 27) {
			break;
		}
	}
	return 0;
}
```

Всероссийское СМИ
«Академия педагогических идей «НОВАЦИЯ»
Свидетельство о регистрации ЭЛ №ФС 77-62011 от 05.06.2015 г.
(выдано Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций)

Сайт: akademnova.ru
e-mail: akademnova@mail.ru

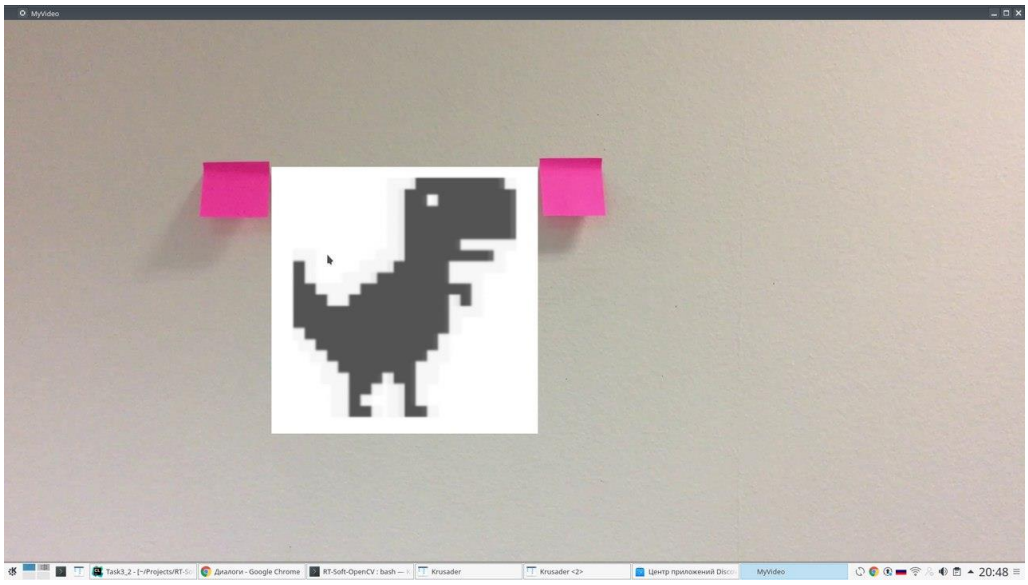

Рисунок 5 – Результат работы программы, отображающей динозавра между стикерами

Список использованной литературы:

1. OpenCV [Электронный ресурс]. – URL: http://robocraft.ru/page/opencv/ (дата обращения 01.10.2017).
2. Преобразование Хафа [Электронный ресурс]. – URL: http://ru-wiki.org/wiki/%D0%9F%D1%80%D0%B5%D0%BE%D0%B1%D1%80%D0%B0%D0%B7%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%A5%D0%B0%D1%84%D0%B0 (дата обращения 01.10.2017).
3. Цифровая обработка сигналов и изображений: лабораторный практикум для студ. спец. I-40 02 01 «Вычислительные машины, системы и сети» всех форм обуч. / М. М. Лукашевич, Р. Х. Садыхов – Минск : БГУИР, 2010. – 35 с. ISBN 978-985-488-598-8

Дата поступления в редакцию: 22.07.2018 г.
Опубликовано: 22.07.2018 г.