**ТАБЕЛЬ**

**ПРОХОЖДЕНИЯ ПРАКТИКИ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Дата | Содержание или наименование проделанной работы | Место работы | Время работы | | Подпись цехового руководителя |
| начало | конец |
| 1 | 2 | 3 | 4 | 5 | 6 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**ТАБЕЛЬ**

**ПРОХОЖДЕНИЯ ПРАКТИКИ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Дата | Содержание или наименование проделанной работы | Место работы | Время работы | | Подпись цехового руководителя |
| начало | конец |
| 1 | 2 | 3 | 4 | 5 | 6 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Лекции 10:00 – 15:00 ч

Самостоятельная работа 10:00 – 15:00 ч

Первое собрание 10:00 – 12:00

Второе собрание 16:00 – 18:00

2.07 – Лекция Использование git и cmake. Работа с видео в OpenCV.

3.07 – Лекция Алгоритмы обработки изображений. Использование шаблонов в C++11/14/17 часть 1.

4.07 – Лекция Алгоритмы обработки SIFT, SURF, ORB. Использование шаблонов в C++11/14/17 часть 2.

5.07 – Лекция Использование нейронных сетей для распознавания изображений. Структуры данных и сложность алгоритмов

6.07 – Лекция Распараллеливание алгоритмов обработки изображений. Потоки C++11/14/17

9.07 – Определение структуры программы, написание каркаса

10.07 – Программирование обработчика кадров видео.

11.07 – Реализация функции отслеживания футбольных ворот

12.07 – Собрание. Подведение промежуточных итогов

13.07 – Изучение возможностей расширенной библиотеки OpenCV

16.07 – Изучение функций отслеживания движений OpenCV

17.07 – Программирование функции отслеживания мяча

18.07 – Реализация функции вывода карты попаданий по воротам. Подготовка презентации для отчета

20.07 – Собрание. Подведение итогов

**Отзыв руководителя практики**

**Московский авиационный институт**

**(национальный исследовательский университет) «МАИ»**

**Факультет №3** **—** «Системы управления, информатика и электроэнергетика»

**Кафедра 307 —** «Технология приборостроения. Конструирование и технология производства средств информационно-вычислительной техники»

Отчет по производственной практике

**Выполнил:**

студент группы 3О-314Б

Медведев Никита

**Принял:**

К.т.н., доцент, заведующий кафедрой 307

Васильев Фёдор Владимирович

Москва 2018

# Задание на практику

Прохождение тренинга по встраиваемым системам компании «РТСофт», состоящего из двух этапов: основного курса лекций по применению языка C++ в приложениях дополненной реальности и выполнение практического задания с применением полученных знаний.

# Процесс прохождения практики

## Курс лекций

Лекции проходили пять дней и содержали теоретическую базу для практического задания.

День 1.  В первой лекции говорилось о системах контроля версий. Были рассмотрены виды таких систем, а также способы работы с системой git: работа с локальными и удаленными репозиториями, их создание, настройка, использование.

Во второй лекции рассказывалось об умных указателях и их применении, а также о библиотеке компьютерного зрения OpenCV.

День 2. В первой лекции речь шла о шаблонном программировании в C++ стандарта 1998, 2011 и 2017 годов.

Во второй лекции говорилось об алгоритмах обработки изображений для выделения линий, сегментации изображений по цвету и гистограммам.

День 3. Продолжилась лекция про шаблонное программирование в C++.

День 4. В первой лекции рассказывалось о нейронных сетях и применении свёрточных нейронных сетей и сетей обратного распространения для обработки изображений.

Во второй лекции говорилось о алгоритмах, их сложности, а также о структурах хранения данных и стандартной библиотеке C++ STL.

День 5. В последний день речь шла о многопоточном программировании.

## Выполнение практического задания

Практическое задание заключалось в написании программы, которая:

- выделяет мяч, игроков, ворота и разметку из видеофайла, содержащего запись серии пенальти в футболе. Это подводящая часть задачи.

- определяет и рисует места попаданий в ворота (на отдельной картинке или на видео). Это основное задание.  
Я начал решать задачу с составления списка функций, которые понадобятся для решения задания. В первую очередь необходимо загрузить видеофайл для его обработки. Затем требуется провести сегментацию картинки, выделив ключевые элементы, такие как ворота, разметка, мяч, игрок. При необходимости – отслеживать эти элементы. На последнем этапе требуется отобразить кадр, а после воспроизведения всего видео нарисовать карту попаданий мячом по воротам.

Также необходимо подобрать видеофайлы для тестирования. Для этого были взяты записи послематчевых серий пенальти нескольких матчей Чемпионата мира по футболу. На этом этапе стало ясно, что нельзя жестко задать координаты ворот, и необходимо находить их в кадре, т.к. в видеонарезке присутствуют кадры трибун, разных ракурсов, а основное положение камеры (рис. 1), из которого показывается удар по воротам в «прямом эфире», не статичен. Постоянство ракурса камеры упрощает задачу, ведь объекты в кадре расположены приблизительно одинаково.



Рисунок 1. Стандартное положение камеры при пенальти в футболе.

Перед написанием функций, я опробовал алгоритмы и преобразования, о которых узнал на лекциях. Начал с цветовых моделей: BGR (blue, green, red; OpenCV использует такую вариацию RGB по умолчанию), HSV (hue, saturation, value – тон, насыщенность, значение) и grayscale (изображение в градациях серого). Выяснилось, что на grayscale–изображении, при использовании фильтра по значению пикселя, отлично видны ворота. Следующими на очереди были алгоритмы выделения линий. Первое, что становилось очевидным – они очень медленные, и применение их на весь кадр замедляло видео в несколько раз.

Здесь началось написание программы. Для оптимизации алгоритма Канни по поиску контуров было найдено решение: при помощи функции поиска по шаблону из библиотеки OpenCV на всем кадре находится участок с воротами, и уже на этом участке ведется поиск линий. Следующее ускорение алгоритма заключается в преобразовании трехканального цветного в серое изображение и дальнейшей фильтрации по значению, которая отсекает лишнее, в итоге алгоритм Канни получилось оптимизировать до применения в реальном времени.

Найденные на предыдущем этапе контуры, несмотря на всю оптимизацию, содержат много лишней информации. Для выделения рамки ворот более точно, при помощи преобразования Хафа среди контуров находятся вертикальные прямые линии, которые имеют ширину и длину, больше заданных, чтобы отсечь вертикальные буквы в рекламе за воротами и другие помехи. Самые высокие и самые низкие точки построенных линий обозначают углы ворот, которые мы ищем (рис.2).

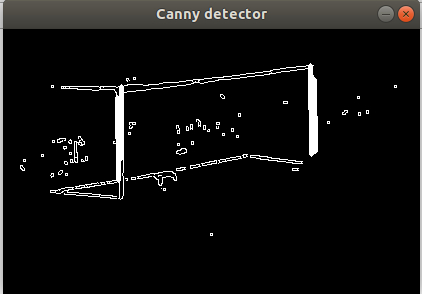


Рисунок 2. Результат работы алгоритма Канни и преобразования Хафа.

Полученный результат дал понять, что для реализации основной задачи – определения точек попадания – не требуется знание разметки поля, и эта подзадача автоматически отпадает.

Для определения кадров с необходимым ракурсом решено сравнивать гистограмму кадра с тестовым изображением, и по разнице определять, это необходимый нам кадр или нет. Из – за того, что тестовое видео снималось на разных стадионах, в разное время суток и при разном освещении, отклонение между гистограммами лежит в большем промежутке, и это приводит к промахам (рис. 3). Решение этой проблемы за время практики найдено не было.



Рисунок 3. Промах в определении кадра.

Последняя функция, которая была реализована, это функция отмечания точек попадания в ворота.

Также принималась попытка реализовать функцию отслеживания мяча. В расширенной библиотеке OpenCV содержатся инструменты для этого, но за время практики не удалось реализовать отслеживание, т.к., в зависимости от алгоритма, мяч или терялся в момент удара за бутсой игрока, или область отслеживания переходила на другой объект (бутсу или шорты игрока) из-за того, что те чётче и мяч движется слишком быстро.

# Итоги практики

Эта практика имела для меня огромную практическую пользу. Я узнал особенности языка C++, такие как шаблонное программирование, научился анализировать библиотеки, разбирать чужой код, и к тому же лучше освоил git и Linux.