





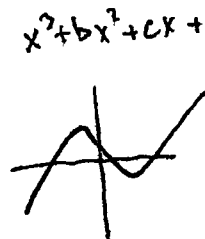
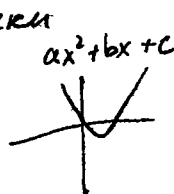


Инвариантная часть

Докажем, что график параболы кубической  $y = x^3 + bx^2 + cx + d$

симметричен относительно некоторой точки

$y' = 3x^2 + 2bx + c$



Точки "излома" параболы - точки, где  $y' = 0$ , т.к.

$y'$  - квадратичная параболы

Точки симметрии квадратичной параболы  $x_0 = \frac{-2b}{6}$

Ответ  ~~$x = \frac{-2b}{6}$~~   $x = \frac{-2b}{6}$  5 баллов

Вариативная часть Блок 3

1 Система распознавания координат игроков с трех камер

Требования на вход Частота кадров 60 к/с

Разрешение достаточное, чтобы в любой точке поля, если номер игрока в видимости камеры его можно распознать

Цвет любой, в т.ч. 2/5

Алгоритм Загрузили 3 видео одного фрагмента матча  
Каждое видео обрабатывается отдельно, на выходе из алгоритма  
анализа видео в JSON со структурой  
Игрок  $\{$

id int (номер игрока)

coords  $\{$  кадр - последовательный номер кадра

x  
y

$\}$

где coords.L - координаты с камеры L



Как распознаем игрока на кадре обращаемся к ML CV

шагую, который

- 1 ~~видеопоток~~ разрезает ~~всю картинку~~ на каждый пиксель на 0 или 255, где 255 - игрок, 0 - фон  
Точность не высокая
- 2 Берем на изображении область пикселей с размером  $\begin{matrix} 2m \\ 0,5m \end{matrix}$  ~~и добавляем ее~~

~~Вносим на поле эту область~~

- 3 Отдаем эту область на определение номера игрока  
номер ищем через opencv  
если номер найден на кадре, то

~~добавляем в массив~~ добавляем  $\text{player}[id][coords] \text{ tuple}(\text{кадр}, x, y)$   
 где  $x, y$  считаются по формуле  
 расстояние -  $\sqrt{x^2 + y^2 + z^2}$   
 с учетом известной позиции камеры  
 расстояние до каждой точки поля с учетом перспективы кадра считается по формуле

1) 15 б.

Если Id на кадре нет, то пропускаем игрока  
с каждой камеры координаты кладем в ~~один массив~~ соответствующий массив <sup>на кадре</sup>

Получили для каждого игрока ~~на поле~~ массив  $(\text{кадр}, x, y)$   
~~на поле~~ кадр считается по номеру кадра инкрементивно, кол-во кадров в каждом видео совпадает координаты могут быть найдены не для каждого кадра в таком случае этот кадр не добавляется в массив coords

на выходе возвращаем Ids



Бланк ответов

2 Алгоритм подсчета метрик  
На входе JSON из прошлого алгоритма, где

```

Players[]
  Player {
    id int
    coords[]
      } cadr int
      x float
      y float
    }
  }

```

Все поля заполнены  
Все игроки, которые  
были найдены

Будем добавлять

~~поля~~ для каждого Player поля  
int mean\_speed - ср скорость за все  
время

mean\_min\_speed [ minute mean speed

Средняя скорость за всю игру считается как средняя всех поминутно  
средних скоростей  $\sum_{n=1}^n \frac{\text{все поминутные}}{n}$  алгебраиз

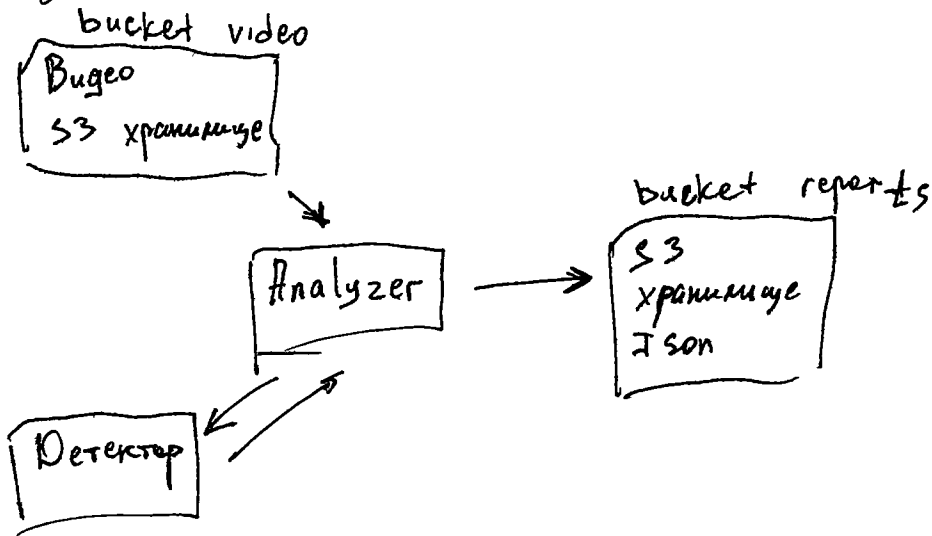
Средняя поминутно так

- 1 Считаем среднюю координату в секунду  
среди всех кадров, с разностью не больше  
(так не для всех кадров есть коорд) во-ти
- 2 Считаем разницу координат для каждой  
секунды  
Получаем секунда м/с  
и для ~~каждой~~ <sup>всех</sup> секунд в пределах одной реальной  
минуты считаем ср скорость минут

На входе Для каждого coords-i считаем свое среднее поминутное

2) 15

# Архитектура системы



## Структура данных

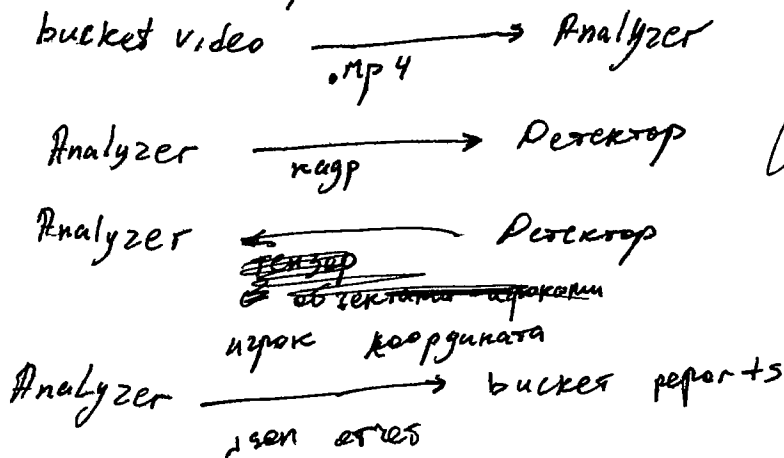
3) 105

- Video - S3 хранилище с bucket video, где хранятся mp4 файлы со структурой:
  - папка матч1
    - камера 1 mp4
    - камера 2 mp4
- S3 хранилище JSON с bucket reports хранятся итоговые JSON отчеты с метрикой с каждого матча структура в формате матч-1 json

## Функционал

- bucket video - хранение видеороликов
- Analyzer - создание отчетов, обработка видео
- Детектор - детекция координат игроков на кадре
- bucket reports - хранение JSON отчетов

## Поток данных



Итого 405