

Пояснения к алгоритмам

Введение

В папке “программы” вы можете видеть 4 файла, а заезда всего 2, это потому, что на каждый заезд у нас есть две программы одна “safe” - безопасная, с низкой скоростью и вторая “fast” - быстрая, с некоторыми дополнениями и оптимизированная под большую скорость.

Квалификация

Здесь будет разобрана только программа “safe”, так как в быстром варианте отличаются только некоторые значения скорости и угла поворота.

Для начала следует взглянуть на поле глазами робота

(фотка с камеры)

На этом фото мы можем видеть поле, телеметрию (текст на экране), а также прямоугольники. Эти прямоугольники — это графическое отображения областей интереса. Они очерчивают границы области поиска тех или иных объектов, например, два красных прямоугольника очерчивают границы областей поиска бортиков, а маленький прямоугольник по центру - область поиска синей и оранжевой линии. В дальнейшем для простоты и краткости область интереса и прямоугольник мы объединим в понятие датчик. То есть на кадре присутствует 3 датчика, 2 датчика бортика и один датчик перекрёстков (перекрёстком называется синяя или оранжевая линия, в зависимости от направления движения).

Датчики

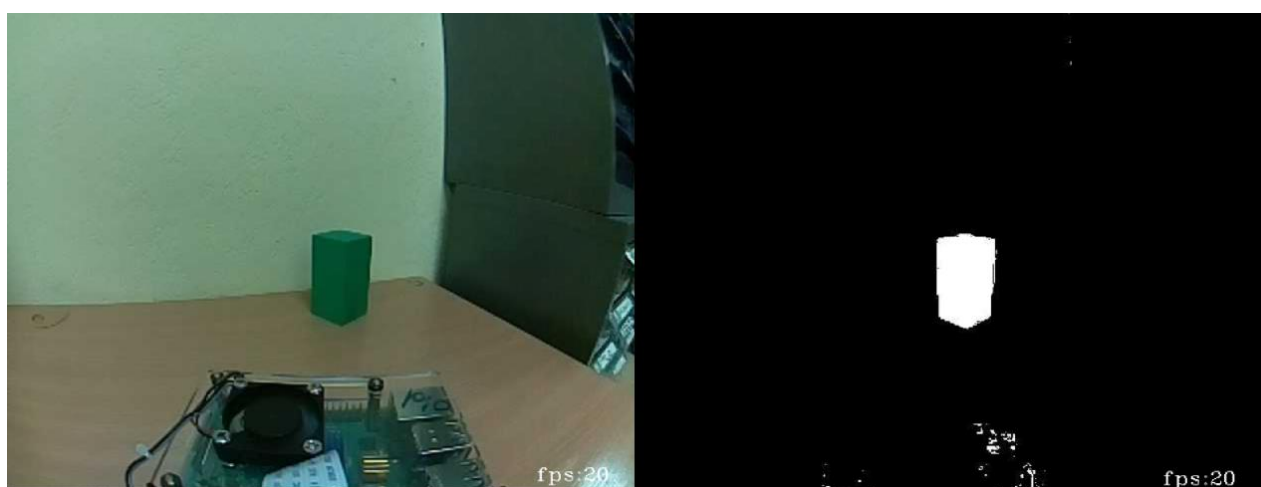
Датчики нужны для того, чтобы определить позицию объекта (бортика, знака, линии) по цвету в модели HSV. Все датчики устроены похожим образом и имеют базовый набор действий:

Скопировать часть кадра --> перевести полученное изображение в цветовую модель HSV --> создать маску --> найти на этой маске контуры --> найти наибольший контур --> передать его координаты.

В этом наборе есть действие, требующее пояснения, таковым является создание маски.

Создание маски — это процесс создания массива нулей и единиц на основе изображения в цветовом пространстве HSV и диапазона значений в той же модели. Алгоритм создания таков:

Если цвет пикселя на исходном изображении находится в данном диапазоне, то пиксель с такими же координатами на маске становится белым, иначе пиксель будет чёрным.



Выше представлена наглядная демонстрация создания маски

На raspberry находится и выполняется основная программа, что подразумевает больше трудностей и особенностей, и соответственно больше алгоритмов. Для удобства разделим все алгоритмы на две группы, а именно на алгоритмы поиска и алгоритмы движения, а имеющиеся группы разделим по заездам на квалификационный и финальный.

Алгоритмы поиска

Вставить картинку

Как мы можем видеть для прохождения квалификационного заезда не нужно много датчиков (датчиками условно называются области интереса, в которых мы что-то ищем, например на двух датчиках по краям экрана мы ищем черный бортик).

На записи квалификации мы видим только три датчика, два датчика для чёрного бортика и один датчик для так называемых перекрёстков (перекрёстками условно называются оранжевые и синие линии). У всех датчиков, несмотря на разные задачи и разные объекты поиска, имеется примерно одинаковый набор действий по нахождению предметов интереса. Классический набор действий включает в себя: выбор части изображения (области интереса), размытие её по Гауссу, перевод изображения в цветовое пространство HSV, получение маски (маской называется черно-белое изображение, полученное посредством изменения цвета всех пикселей изображения, все пиксели, попавшие в установленный диапазон, становятся белыми, а все остальные становятся чёрными), поиск контуров на этой маске, и последующее нахождение координат этих контуров. Полученные координаты контуров обычно и являются выходными данными датчиков.

Теперь разберём датчики в отдельности, начнём с датчиков чёрного бортика (далее датчики линии).

Датчики линии занимаются поиском черного цвета и в набор их действий входят все вышеописанные операции. Левый датчик определяет самое правое положение черного цвета, а правый датчик определяет самое левое положение чёрного цвета. Исходя из показаний этих двух датчиков считается управляющее воздействие, но к нему мы вернёмся в разделе “Алгоритмы движения”.

Датчики линии постоянно видят свой объект интереса, в отличие от датчика перекрёстка. Датчик перекрёстка, как следует из названия, ищет перекрёстки и считает их количество. Но перекрёсток находится на экране не один кадр, следовательно возможны ложные срабатывания и некорректное подсчитывание кругов. Исходя из этого мы написали алгоритм, решающий эти проблемы. При обнаружении перекрёстка мы запускаем таймер, и регистрируем перекрёсток только если с предыдущего перекрёстка прошло какое-то время, зависящее от скорости.

Также в алгоритм работы всех датчиков включено ограничение по минимальной площади контура.

В программе для финального заезда добавляются ещё два датчика, для поиска знаков. Они отличаются только тем, что они ищут разные цвета.

Алгоритмы движения

Самый основной алгоритм движения, который используется в каждой из программ — ПД-регулятор.

ПД-регулятор — это алгоритм, высчитывающий управляющее воздействие, исходя из показаний датчиков. В квалификации мы используем один и тот же регулятор на протяжении всего заезда, но бывают ситуации в которых регулятор не помогает, и для таких ситуаций нужны особые алгоритмы. Например: при потере одного датчика (потеря датчика — ситуация, в которой показания датчика равны нулю) мы должны повернуть в направлении этого датчика, то есть приблизиться к бортику, от которого мы отделились. А если потеряны оба датчика, робот должен произвести поворот в направлении движения, которое мы определяем по цвету первого перекрёстка.

В программе для финального заезда добавляются ещё два ПД-регулятора, для зеленого и красного знаков.