

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Обзор существующих алгоритмов планирования траекторий движения БЛА в среде с препятствиями	7
1.1 Обзор используемых сенсоров	7
1.1.1 Видео-камера	8
1.1.2 Лидар	10
1.1.3 Ультразвуковой датчик	11
1.1.4 Радар	11
1.2 Алгоритмы облета препятствий с использованием датчиков расстояния	12
1.2.1 Алгоритм Tomoyuki Mori и Sebastian Scherer	12
1.2.2 Алгоритм Chelsea Sabo и Alexander Simpson	16
1.2.3 А. Barry и R. Tedrak	19
1.3 Алгоритмы облета препятствий с использованием датчиков расстояния	20
1.3.1 Алгоритм Raimundo и Sebastião	20
1.3.2 Алгоритм Stefan Hrabar	22
1.3.3 Алгоритм J. Saunders, B. Call, A. Curtis и R. Beardr	24
1.4 Вывод по главе	25
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	27

ВВЕДЕНИЕ

Способность птиц замечать и маневрировать между препятствиями в окрестностях городской и дикой местности, является объектом множества исследований. В следствии этого произошло бурное развитие летательных аппаратов.

Актуальность работы определяется современными тенденциями роста науки и техники. В течении последних двенадцати лет малые беспилотные летательные средства прошли путь от первого прототипа, до самого востребованного устройства во многих областях деятельности. Беспилотное летательное средства (БЛА) — летательный аппарат без экипажа на борту. [1] Благодаря способности работать в отдаленных и опасных ситуациях и возможности автономного управления, БЛА можно встретить в разных сферах применения, например, таких как: геодезия и картография, спасательные операции при стихийных бедствиях, сбор данных из труднодоступных районов, разведка, видеосъемка и фотосъемка, транспортировка. В большинстве применений основной задачей БЛА является перемещения из начального пункта в конечный. Автоматизирование процесса перестройки маршрута в зависимости от окружающей местности, может сократить нагрузку на оператора и позволит ему управлять большим количеством устройств одновременно. Также человеческий фактор будет сведен к минимуму.

Беспилотные летательные устройства могут обладать разной степенью автономности – автоматические, полуавтоматические, управляемые дистанционно. БЛА управляемые дистанционно требуют постоянного управления пилота. Полуавтоматические БЛА, требуют кратковременных команд управления от пилота, в трудно проходимых местах. Автоматические БЛА, полностью автономны, управление от пилота не требуется. Одной из наиболее сложных проблем в области автономных воздушных транспортных средств является разработка надежной системы обнаружения и предотвращения столкновения с препятствиями в режиме реального времени. Решение данной проблемы,

позволяет уменьшить количество информации необходимой БЛА для решения поставленной перед ним задачи, вследствие этого персонал, обслуживающий БЛА, может быть менее квалифицированным. Так же решение задачи позволит использовать БЛА в неисследованной ранее местности.

В данной работе рассмотрим и сравним различные способы решения задачи облета препятствий в реальном времени. Объектом управления будет являться мультикоптер. Мультикоптер — подвид БЛА, построенное по вертолетной схеме с тремя и более несущими винтами. Сравнение будет проводится по следующим критериям — надежность, количество степеней свободы, необходимые датчики, сложность реализации.

1 Обзор существующих алгоритмов планирования траекторий движения БЛА в среде с препятствиями

Все рассмотренные ниже алгоритмы можно разделить на две группы:

- алгоритмы, использующие для распознавания препятствий видеокамеру;
- алгоритмы, использующие для распознавания препятствий датчики расстояния;

У каждой группы есть свои достоинства и недостатки. Из-за небольшой грузоподъемности малых БЛА использование большого числа навесных устройств невозможно. При использовании дальномеров происходит измерение расстояния от наблюдателя до малой окрестности вблизи точки на объекте. Иногда это не достаточно информативно. В отличие от дальномеров, видеокамера, дает большой поток информации описывающий окружающую обстановку. Но этот поток необходимо обработать, для детектирования объектов, что несет увеличение сложности алгоритма, а так же необходимость в более производительном вычислительном устройстве.

Объединенные системы — 3D-камеры не имеют указанных недостатков. Например RealSense D435 от компании intel или Kinect от компании Microsoft. Данные системы позволяют детектировать препятствия, получать видео поток, расстояние до объектов. Но с усложнением измерительных устройств растет и их вес, размер и необходимые для обработки данных вычислительные ресурсы.

Далее рассмотрим используемые сенсоры подробнее.

1.1 Обзор используемых сенсоров

1.1.1 Видео-камера

Видео-камера — устройство для съемки объектов, путем преобразования оптического изображения в электрический сигнал. Простейшая камера состоит из:

- объектив;
- матрица;
- схема для обработки видео сигнала.

Объектив служит для формирования изображения на матрице. Состоит из системы линз, показан на рисунке 1.1.

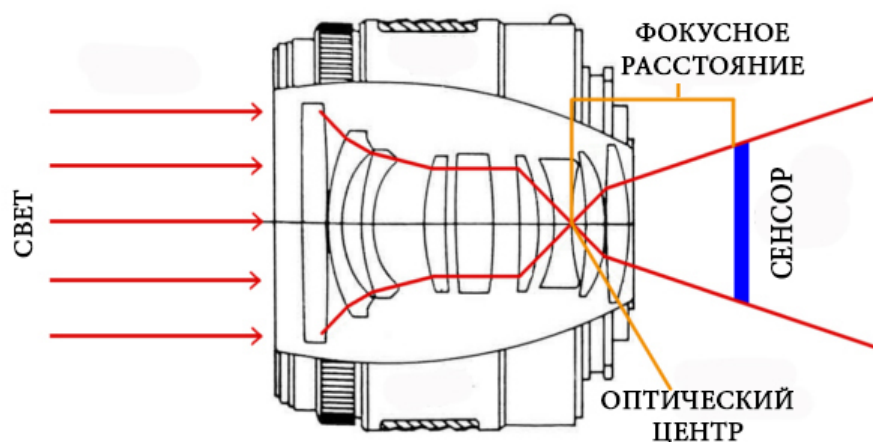


Рисунок 1.1 – Модель объектива.

Основной характеристикой объектива является фокусное расстояние. Фокусное расстояние это расстояние от оптического центра объектива до точки фокусировки, то есть до матрицы. Чем больше фокусное расстояние, тем дальше может находиться объект съемки, но меньше угол видимого объективом охвата.

Для преобразования оптической информации в электрический сигнал для дальнейшей обработке на вычислительном устройстве необходима матрица. Существует два основных типа матриц:

- ПЗС матрицы;
- КМОП матрицы.

Матрицы состоят из светочувствительных пикселей, задача которых передать электронный сигнал соответствующий количеству света попадаемого на него. В ПЗС матрицах заряды сдвигаются по матрице строчками сверху

вниз, перед выходом из сенсора, заряды усиливаются, в итоге получается аналоговый сигнал с изменяющимся напряжением зависящим от количества света попавшего на пиксель. ПЗС матрица представлена на рисунке 1.2. Для преобразования в байты используют аналого-цифровой преобразователь.

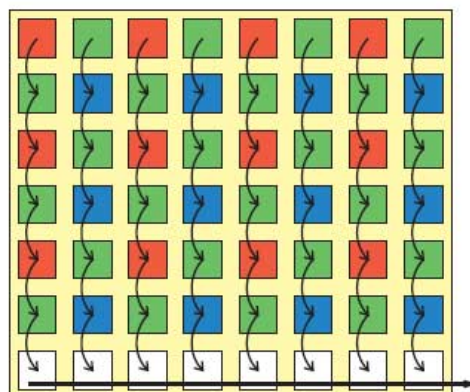


Рисунок 1.2 – Модель ПЗС матрицы.

Преимущество таких матриц — минимальное количество шумов. В КМОП матрицах преобразование сигнала происходит прямо на пикселе. Соответственно считывание происходит сразу с нужного пикселя, что ускоряет работу таких матриц. КМОП матрица представлена на рисунке 1.3

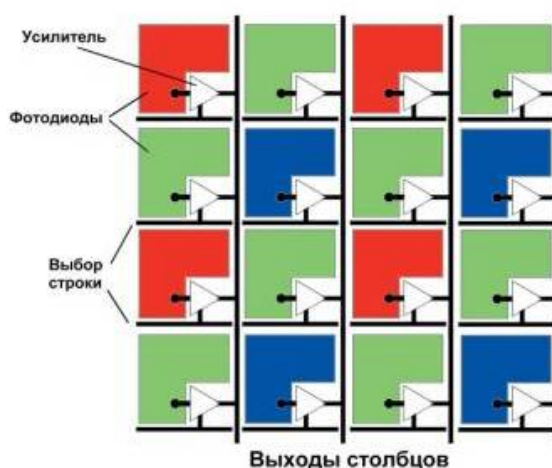


Рисунок 1.3 – Модель КМОП матрицы.

Схема для обработки видео сигнала необходима для управления режимом считывания, формирования и усиления видеосигнала.[13]

1.1.2 Лидар

Лидар — это измерительная технология, основанная на точном измерении временной задержки между передачей импульсного оптического лазерного светового сигнала и его приема. Лидар как датчик — это активный оптический сенсор, испускающий лазерные лучи в сторону цели во время движения транспортного средства по специальным съемочным маршрутам. Он работает с двумя типами световых сигналов: опорный сигнал, генерируется самим датчиком и отраженный сигнал. Для определения расстояния необходима временная задержка между этими сигналами и скорость света. Расчет расстояния происходит по следующей формуле:

$$R = c * \frac{\Delta t}{2}$$

Где R — расстояние до объекта;

c — скорость свет, равная 299792458 метров в секунду;

Δt — это временная задержка между импульсами.

Обычно лидары работают на ультрафиолетовых и инфракрасных волнах, что соответствует диапазонам длин волн 100-400 нанометров и 1500-2000 нанометров. [9] Диапазон длин волн используемых лидаром представлен на рисунке 1.4.

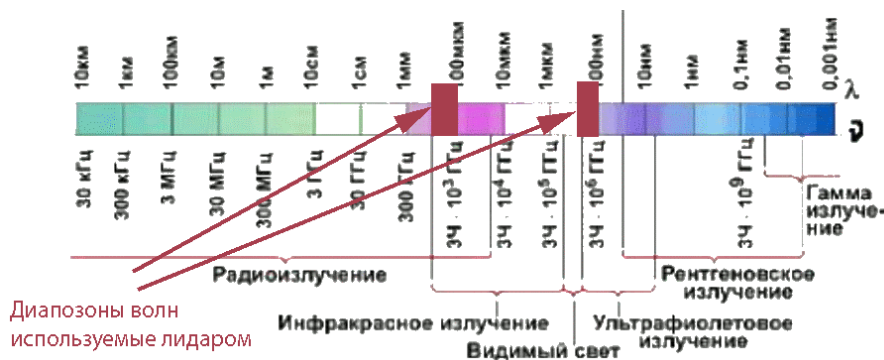


Рисунок 1.4 – Шкала электромагнитных волн.

1.1.3 Ультразвуковой датчик

Ультразвуковой датчик состоит из источника ультразвукового импульса и приемника. Принцип работы заключается в измерении времени между отправленным ультразвуковым импульсом и отраженным от объекта. На основе этого времени и скорости звука в среде распространения импульса вычисляется расстояние до объекта. Между измерениями должен быть временной промежуток, для того чтобы эхо от предыдущего импульса исчезло. Главным преимуществом ультразвукового датчика — возможность детектирования прозрачных объектов, например застекленных окон.

1.1.4 Радар

Радар — это технология измерения расстояния, основанная на радиоволнах. Принцип работы такой-же как у ультразвукового датчика и лидара, который состоит в обнаружении объектов на основе отраженных излучаемых сигналов. Состоит радар из:

- антенна;
- передатчик;
- приемник.

Передатчик генерирует и посылает короткий, высокой интенсивности, пакет высокочастотных радиоволн. Антенна, после отражения этой волны от объекта, классифицирует волну как радиозхо-сигнал. Приемник усиливает этот сигнал. Измеряется время возврата эхо-сигнала, а также его Доплеровский сдвиг. Расчета расстояния до объекта, происходит как и в ультразвуковом датчике и лидаре. Доплеровский сдвиг необходим для расчета скорости объекта.

Эффект Доплера — явление изменения частоты и, соответственно длины волны излучения, воспринимаемое наблюдателем, вследствие движения источника излучения или движения наблюдателя. [14] Иллюстрация к эффекту Доплера приведена на рисунке /refdopler. [8]

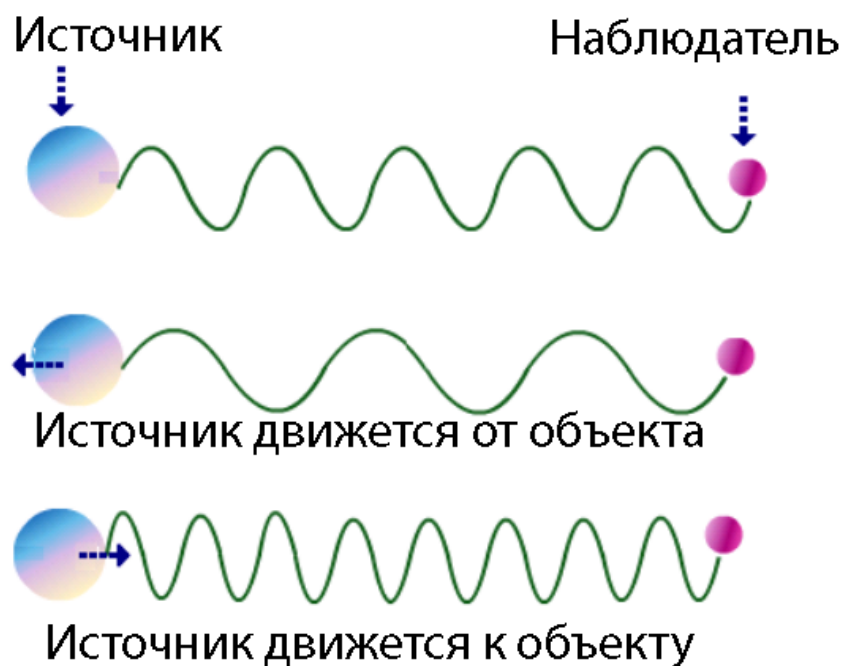


Рисунок 1.5 – Эффект Доплера.

Далее рассмотрим алгоритмы облета препятствий с использованием видео-камер.

1.2 Алгоритмы облета препятствий с использованием датчиков расстояния

1.2.1 Алгоритм Tomoyuki Mori и Sebastian Scherer

В своем исследовании Гибсон [3] смог доказать, что увеличение размера объекта человеком воспринимается как приближение объекта, а не как роста физических размеров объекта. Эта идея используется в основе алгоритма.

Сначала необходимо выделить из видео потока препятствия. Для этого на двух последовательных кадрах, взятых с задержкой, находятся особые точки методом SURF.

Метод SURF решает две задачи – поиск особых точек изображения и создание их дескрипторов (описательного элемента, инвариантного к изменению масштаба и поворота). Кроме того, сам поиск ключевых точек тоже должен

обладать инвариантностью, т.е. повернутый объект сцены должен обладать тем же набором ключевых точек, что и образец. Метод использует матрицу Гессе для нахождения особых точек. Определитель этой матрицы (гессиан) достигает максимума в точках максимального изменения градиента яркости. Матрица Гессе:

$$H(f) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & ; \frac{\partial^2 f}{\partial x_1 \partial x_2} & ; \dots & ; \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & ; \frac{\partial^2 f}{\partial x_2^2} & ; \dots & ; \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & ; \vdots & ; \ddots & ; \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & ; \frac{\partial^2 f}{\partial x_n \partial x_2} & ; \dots & ; \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Где $f(x,y)$ — функция изменения градиента яркости.

Определитель матрицы Гессе инвариантен относительно поворота, но не инвариантен к масштабу. Поэтому метод SURF использует разномасштабные фильтры для нахождения гессианов. Для каждой ключевой точки считается градиент и масштаб. Градиент в точке вычисляется с помощью фильтров Хаара. Фильтра Хаара показан на рисунке 1.6, черные области имеют значение 1 белые ”0”.

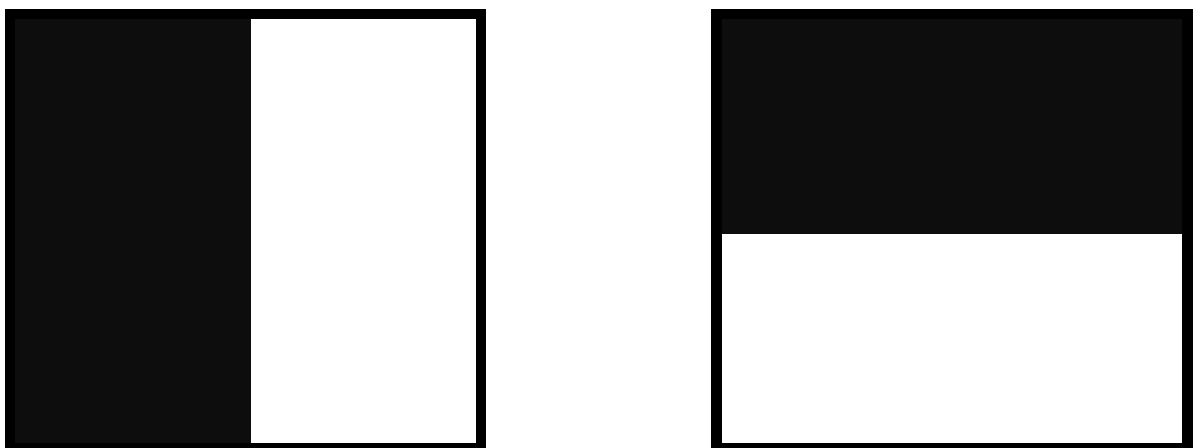


Рисунок 1.6 – Фильтры Хаара.

После нахождения особых точек, формируются их дескрипторы. [4] Найденные особые точки представлены на рисунке 1.7.

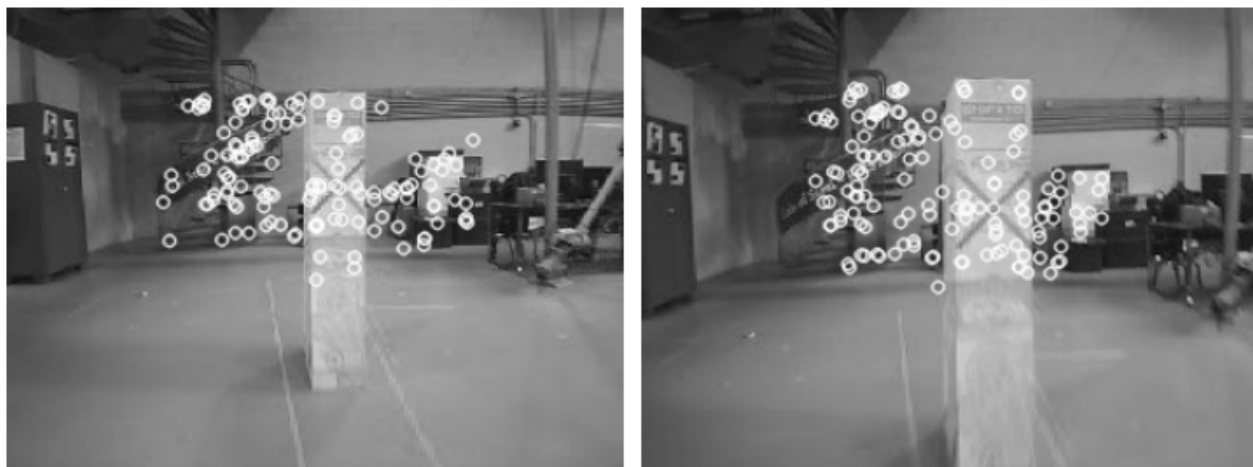


Рисунок 1.7 – Найденные особые точки на двух последовательных изображениях.

На двух последовательных изображениях ищутся соответствие между особыми точкам, показано на рисунке 1.8.

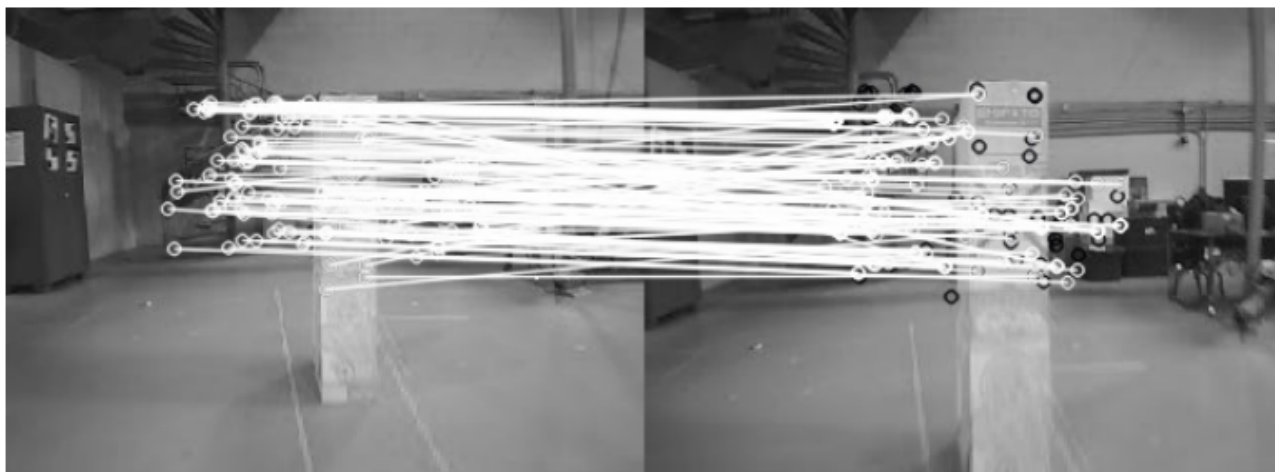


Рисунок 1.8 – Найденные соответствия между особыми точками.

Далее полученные соответствия фильтруются, от ошибочных соответствий, смотри рисунок 1.9 и выделяются области в которых произошло увеличение размера, показано на рисунке 1.10.

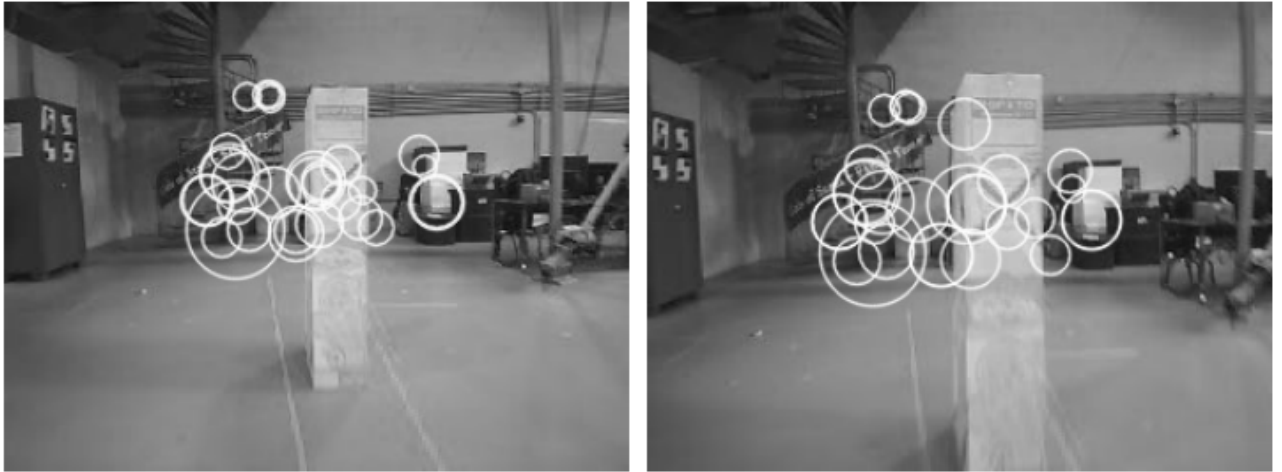


Рисунок 1.9 – Отфильтрованные особые точки, с выделенными увеличившимися участками изображения.



Рисунок 1.10 – Пример изображения, с выделенным красным препятствием и сделанным выбором, по облету его облету.

Размер увеличивается в соответствии с экспериментально вычисленной зависимостью величины объекта от дистанции до него. По этому расстоянию и текущей скорости вычисляется время до столкновения. Маневрирование ведется в горизонтальной плоскости. Решение по облету препятствия принимается в пользу той стороны, где путь будет меньше. [2]

Данный алгоритм позволяет облетать фронтальные препятствия используя камеру с низким разрешением. Алгоритму не требуется априорная информация о препятствиях. Главный недостаток алгоритма, это маневрирование только в горизонтальной плоскости.

1.2.2 Алгоритм Chelsea Sabo и Alexander Simpson

Разработанное решение основывается на ряде биологических механизмов, основным из которых является модель — ”найти брешь представленную Лином [?] Эта модель была разработана для объяснения траекторий полета птиц, пролетающих через поля препятствий. Исследователи обнаружили, что птицы направляются в направлении центра наибольшего зазора в своем поле оптического потока. Интерпретация этой модели показала потенциал для удовлетворения многих требований текущих исследований, особенно в отношении простоты и скорости. Для каждого кадра камеры алгоритм оценил бы подходящее направление рулевого управления, идентифицируя области с низким оптическим потоком, поскольку эти области обычно представляют промежутки или местоположение более отдаленных объектов.

Алгоритм облета препятствий делится на три части – вычисление оптического потока, сегментация, определение направления облета препятствия.

Оптический поток — это изображение видимого движения объектов, поверхностей или краев сцены, получаемое в результате перемещения наблюдателя (глаз или камеры) относительно сцены. Алгоритмы, основанные на оптическом потоке, — такие как регистрация движения, сегментация объектов, кодирование движений и подсчет диспаратета в стерео, — используют это движение объектов, поверхностей и краев. [7] Для вычисления оптического потока берется два кадра видео и генерируется поле, которое содержит массив векторов, отображающих величину оптического потока в каждой точке кадра. В открытой библиотеке OpenCV, уже существует функция вычисляющая оптический поток — Farnback. Вычисление происходит с подбором нескольких параметров. Эти параметры подбираются в зависимости от желаемой скорости работы и точности вычисления потока. На рисунке 1.12 приведены два полученных оптических потока с разными параметрами. Исходное изображение на рисунке 1.11.



Рисунок 1.11 – Исходное изображение.

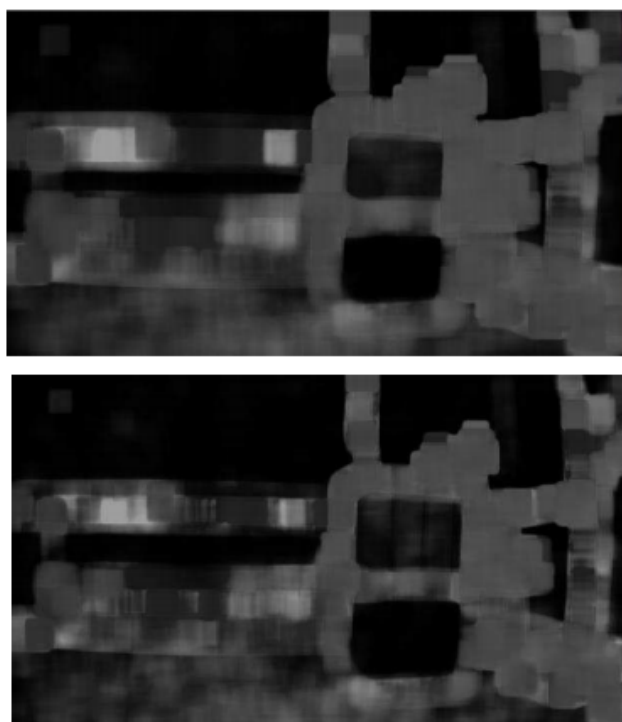


Рисунок 1.12 – Оптический поток вычисленный по рисунку 1.11.

После вычисления оптического потока для каждого кадра, разделим потоки на сегменты. Это разделение поможет выделить потенциальное направление для облета препятствия. Размер и форма сегментов имеет сильное влияние на работу алгоритма. Сегмент должен давать достаточно информации об окружающей среде, но также сегмент должен быть достаточно большим чтобы сквозь него мог пролететь БЛА. На рисунках 1.13 и 1.14 показаны оригинальные кадры и полученные сегментации: 4x2, 8x4, 16x8, 32x16, 64x32, 128x64.



Рисунок 1.13 – Пример сегментации оптического потока.

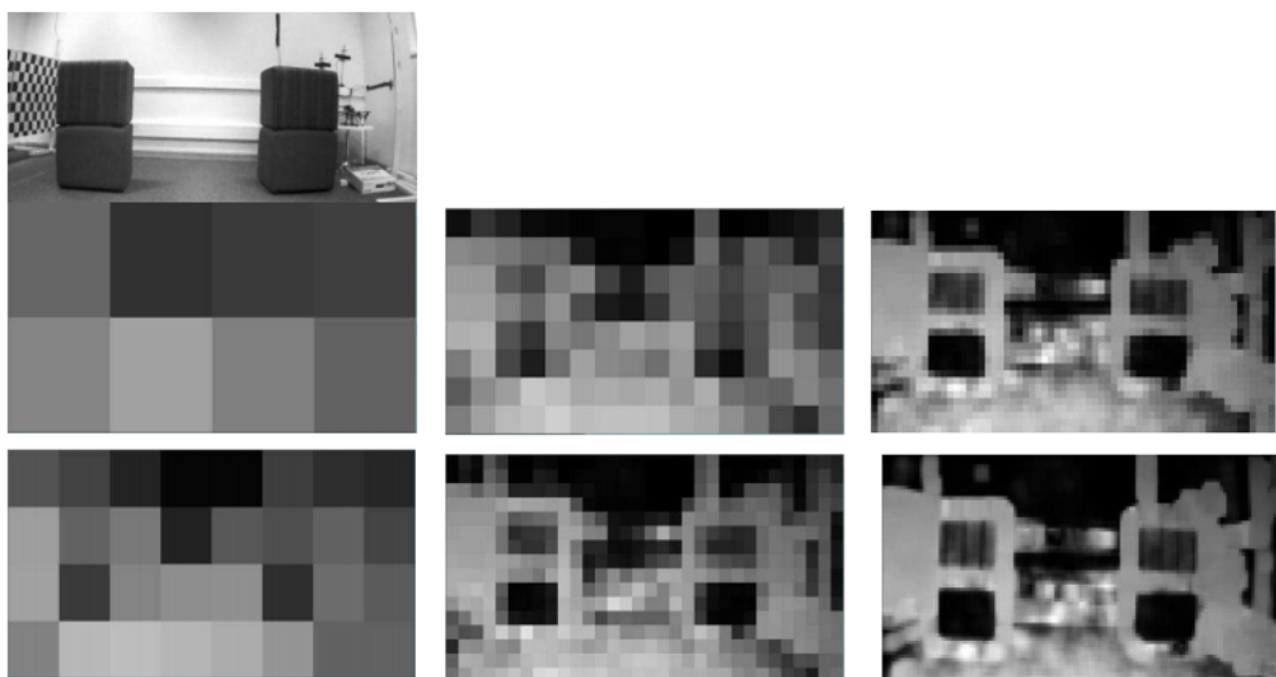


Рисунок 1.14 – Пример сегментации оптического потока.

Последним шагом алгоритма является определение направления облета препятствия, то есть нахождения свободных областей. Такие области можно найти на сегментированных изображениях. Это сегменты с наименьшей величиной оптического потока, то есть более темные участки. Способность алго-

ритма точно и надежно идентифицировать направление зазора сильно зависит от размера используемых сегментов. [7]

Достоинством метода является возможность выбора между скоростью работы и точностью, подбором параметров. Недостатком является сложность подбора размера сегмента для выполнения комплексной задачи.

1.2.3 А. Barry и R. Tedrak

Алгоритм разработан для облета препятствий на большой скорости. Главная идея метода — уменьшить количество вычислений необходимых для детектирования препятствий. Для его реализации на БЛА, необходимо установить две видео-камеры.

На изображениях взятых с двух видео-камер, с помощью метода SAD производится детектирование блоков размером 5×5 с похожими пикселями. SAD - мера схожести между двумя изображениями, получаемая при вычислении абсолютных разностей между каждым пикселем одного изображения и очередным пикселем во втором. Используя найденные блоки и оператор Лапласа высчитаем значение всех блоков, по нему будет происходить пороговая фильтрация.

$$S = \frac{\sum_{i=0}^{5 \times 5} |p(i)_{left} - p(i)_{right}|}{\sum_{i=0}^{5 \times 5} L(p(i)_{left}) + L(p(i)_{right})}$$

В полученной выборке находится большое количество точек горизонтальных граней. В большинстве случаев зафиксированными объектами является линия горизонта. Так как на открытой местности преобладают вертикальные объекты, то горизонтальные можно игнорировать. Для фильтрации горизонтальных объектов, необходимо удалить все объекты, которые на изображениях с правой и левой камеры имеют различия. Пример работы фильтра показан на рисунке 1.15. После обнаружения препятствий, необходимо их облететь по оптимальной траектории. [12]

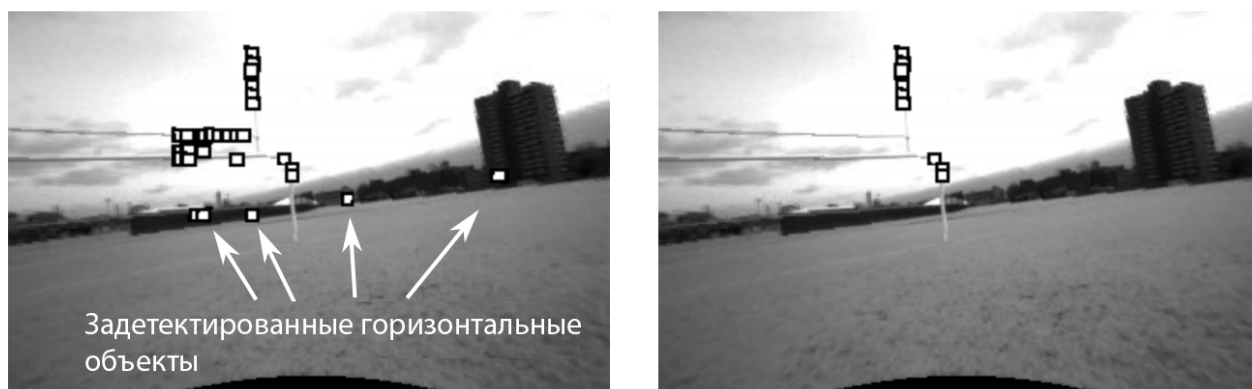


Рисунок 1.15 – Изображение полученное с камеры без фильтра (слева) и с фильтром горизонтальной инвариантности (справа)

Данный алгоритм позволяет не тратя больших вычислительных мощностей, детектировать препятствия. Это позволяет увеличить скорость полета БЛА, без изменения надежности облета препятствий. Алгоритм подходит для полетов в разряженной от препятствий местности, при большом количестве препятствий алгоритм будет отфильтровывать необходимую для безопасного облета информацию.

1.3 Алгоритмы облета препятствий с использованием датчиков расстояния

Далее будут представлены алгоритмы облета препятствий с использованием датчиков расстояния.

1.3.1 Алгоритм Raimundo и Sebastião

Данный алгоритм для измерения расстояния использует ультразвуковой дальномер.

Алгоритм имеет два режима – «Остановка» и «Облети и продолжи». Режим «Остановка» предназначен для управляемых полетов, а режим «Облети и продолжи» для полностью автономных задач.

Режим "Остановка" работает, только при ручном управлении. Если показания Лидара, говорят о нахождении препятствия по пути в радиусе 7 метров, то БЛА останавливается. Если из-за большой скорости БЛА затормозил на расстоянии меньшем 7 метров до препятствия, то ручной режим отключается и мультикоптер в автономном режиме летит назад, пока расстояние не станет больше 7 метров. После этого управление опять передается пилоту. Блок-схема режима представлена на рисунке 1.16.

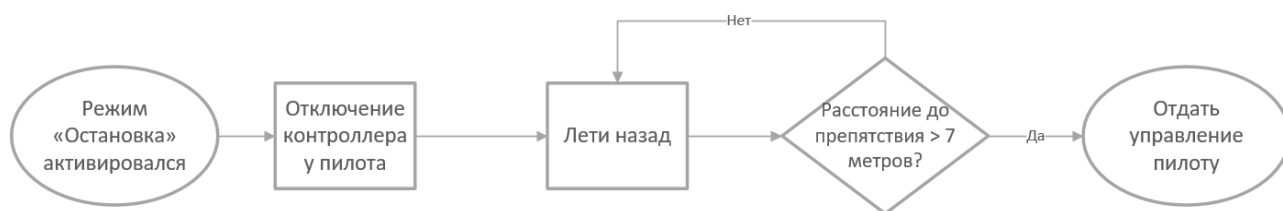


Рисунок 1.16 – Блок-схема режима "Остановка".

Режим "Облети и продолжи" используется для полностью автономных задач. Данный режим гарантирует не только облет препятствий, но и возобновление движения к желаемой цели. Первоначальный путь строится по GPS, в виде контрольных точек. Включении режима «Облети и продолжи» происходит, если на пути появляется препятствие. Сначала алгоритм останавливает начальную миссию, так как на пути появилось препятствие. Далее необходимо запомнить следующую путевую точку, для того чтобы после облета полететь в нужную сторону. После этого БЛА начинает облет препятствия влево или вправо, направление выбирается случайно, так как из датчиков есть только Лидар, то для измерения ширины объекта, необходимо потратить большое количество времени. По малому отклонению в каждую из сторон мы можем понять в какую лететь более перспективно. Если в одну из сторон расстояние до объекта увеличилось и входит в диапазон от 7 до 10 метров, значит сторона выбрана правильно. Расстояние до объекта более 10 метров означает конец облета. Блок-схема режима представлена на рисунке 1.17. [8]

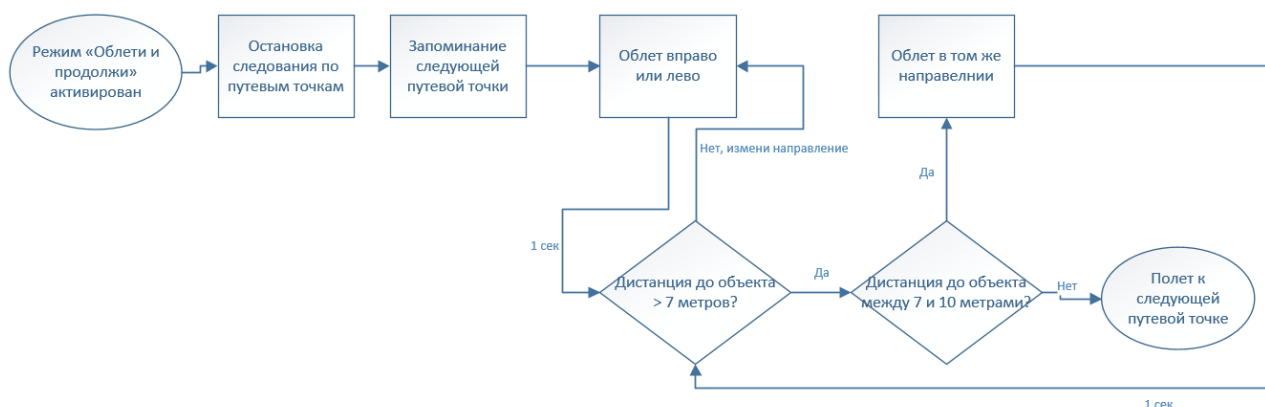


Рисунок 1.17 – Блок-схема режима ”Облети и продолжи”.

Алгоритм облета препятствий работает только в определенных случаях. Подходящий случай – это, если объекты имеют малый размер либо располагаются как показано на рисунке 1.18 — слева, то есть под углом к БЛА. В случае же рисунков 1.18 — посередине и правого, мультикоптер следуя данному алгоритму не облетит препятствия.

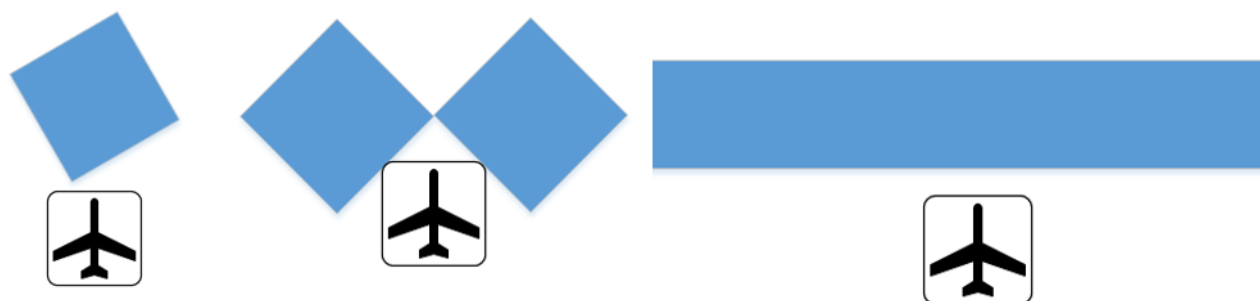


Рисунок 1.18 – Возможные расположения БЛА и препятствий.

1.3.2 Алгоритм Stefan Hrabar

Для реализации данного алгоритма необходим мультикоптер с установленной 3D-камерой, которая позволяет получать облако точек в пространстве. После программной обработки этих точек, получается 3Д-карта местности.

3Д-карта местности используется для представления среды и постоянно обновляется по мере того, как датчики дальности генерируют новые данные.

Данный метод позволяет искать точки облета во всех направлениях, как по горизонтали, так и по вертикали. Также возможно легкое наложение условий для облета препятствий путем рассмотрения только части эллипса. Недостатком является невозможность облета препятствий больших, чем размер самого большого эллипса.

1.3.3 Алгоритм J. Saunders, B. Call, A. Curtis и R. Beardr

Данный алгоритм разработан для беспилотных летательных дронов, построенных по типу самолета. Главным конструкторским отличием от мультикоптера описанного ранее — это наличие крыла, которое создает подъемную силу под действием потоков воздуха. Данный алгоритм справедлив и для мультикоптера. Измерять расстояние до объектов будет лидар.

Сканирующий лидар позволяет узнать расстояние до объектов сразу в нескольких направлениях. На рисунке 1.20(б) красными крестами обозначены замеченные препятствия. Далее удалим из обработки препятствия, которые находятся позади БЛА, так как они не представляют угрозы. Создадим равнобедренные треугольники, середина основания которых, будет совпадать с точкой, замеченного препятствия. Линия высоты этого треугольника должна быть параллельна линии проходящей через следующую путевую точку (на рисунке 1.20 отмечена синим) и текущую позицию БЛА. Длина основания - это расстояние между точкой препятствия и путевой точкой. Длина боковой стороны выбирается исходя из пространства необходимого для маневра БЛА. Сгенерированные треугольники показаны на рисунке 1.20(г). Далее линии треугольников, которые пересекаются с препятствием удаляются, рисунок ???. Оставшиеся линии, это возможные пути для облета препятствий. Выбирается один, с наименьшей длиной пути — рисунок 1.20. [11]

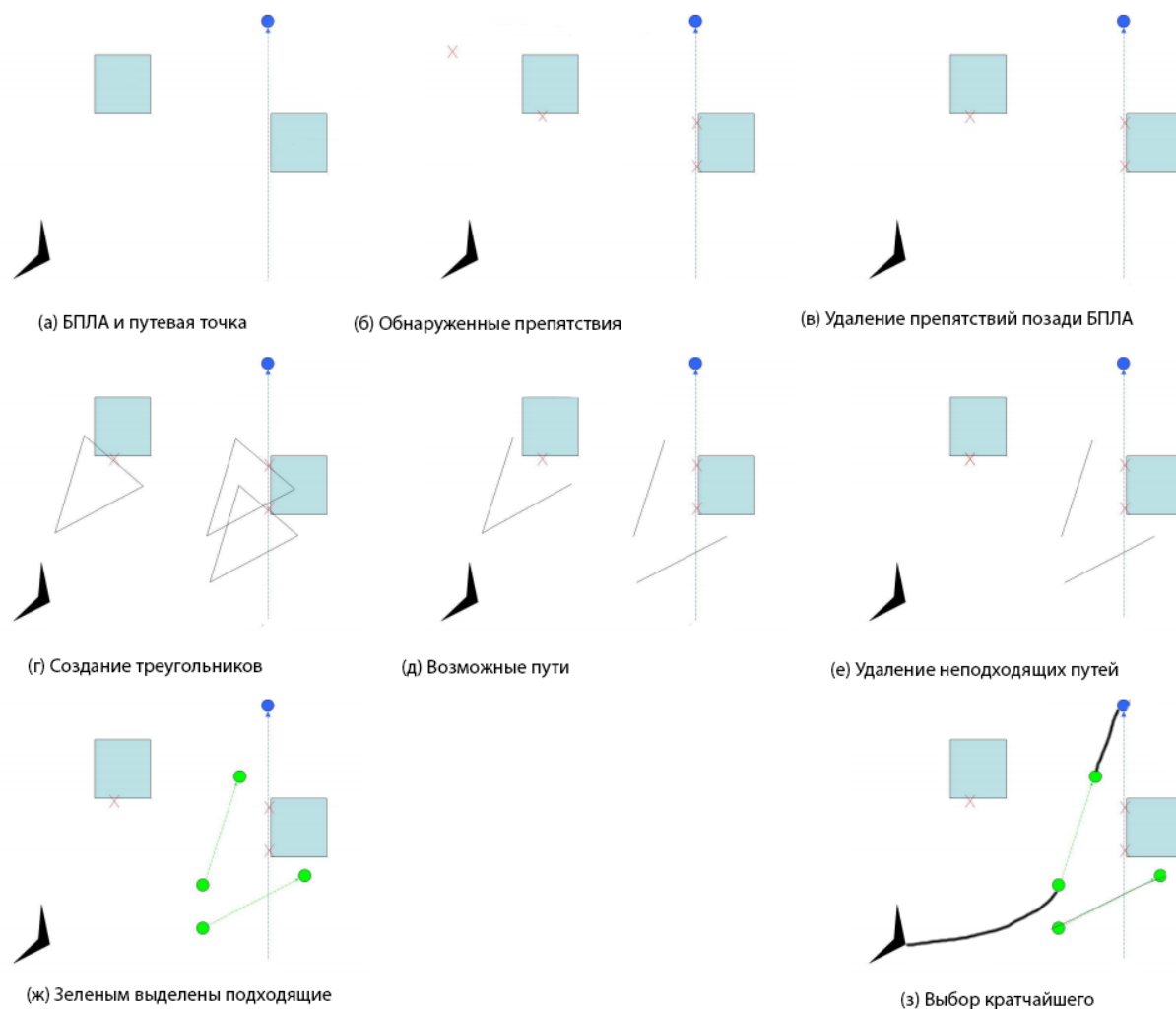


Рисунок 1.20 – Работа алгоритма.

Данный алгоритм позволяет найти не только точку облета, но и небольшой участок облета, что позволяет проходить по более оптимальным маршрутам. Так-же в ходе работы алгоритма, получается несколько возможных путей. При небольшой модификации алгоритма, появляется вариативность маршрута и возможность настройки выбор пути для другого условия. Минусом алгоритма является облет препятствий только в горизонтальной плоскости и сделанное допущение, что сканирующий дальномер увидел достаточно точек на препятствии.

1.4 Вывод по главе

Сравнение описанных в данной главе методов представлено в таблице-1.1.

Таблица 1.1 – Сравнительная таблица методов облета препятствий в реальном времени

Метод	Датчик	Степеней свободы	Надежность	Сложность реализации
Scherer	Камера	2	Средняя	Высокая
Simpson	Камера	3	Средняя	Средняя
Barry	Камера	2	Средняя	Высокая
Raimundo	Ультразвуковой дальномер	2	Низкая	Низкая
Hrabar	3D-камера	3	Высокая	Средняя
Saunders	Лидар	2	Высокая	Высокая

Анализируя данную таблицу приходим к выводу, что наиболее подходящий для данной задачи алгоритм — это алгоритм разработанный Hrabar [10]. Данный алгоритм имеет возможность облетать препятствия задействуя три направления движения, что дает преимущество в оптимальности траектории перед методами разработанными Scherer, Barry, Raimundo, Saunders. 3D-камера позволяет БЛА получать большой объем информации об окружающей среде, даже при плохой освещенности, что повышает надежность по сравнению с использованием видео-камер. Сложность реализации является средней, так как, вычисление расстояний до объектов берет на себя 3D-камера, в отличие от видео-камеры. Также возможно легкое наложение условий для облета препятствий путем рассмотрения только части эллипса.

Недостатком является невозможность облета препятствий больших, чем размер самого большого эллипса. Еще одним минусом является 3D-камера, она имеет большее энергопотребление и цену по сравнению со всеми остальными сенсорами.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Г. П. Свищёв, — Авиация: Энциклопедия М.: Большая Российская энциклопедия, — 736 с. 1994.
2. M. Tomoyuki, S. Scherer, “First Results in Detecting and Avoiding Frontal Obstacles from a Monocular Camera for Micro Unmanned Aerial Vehicles”, Proceedings IEEE International Conference on Robotics and Automation, 8 стр., 2013.
3. J. J. Gibson, “The Ecological Approach To Visual Perception”, 1979.
4. А. С. Иргит, А.В. Максимов ”Применение метода SURF в системах контроля и управления доступом на основе биометрических технологий г. Томск, ТУСУР, РТФ, 2012.
5. Chelsea Sabo, Alexander Simpson ”MAV Obstacle Avoidance using Biomimetic Algorithms Conference: AIAA SciTech 2016, At San Diego, 2016.
6. H.-T. Lin, I. Ros and A. Biewener, ”Through the eyes of a bird: modelling visually guided obstacle flight,”J R Soc Interface, vol. 11, 2014.
7. Оптический поток [Электронный ресурс].— https://ru.wikipedia.org/wiki/Оптический_поток. (Дата обращения: 17.03.2018)
8. A. Raimundo, PhD. P. Sebastião, ”Autonomous Obstacle Collision Avoidance System for UAVs in Rescue Operations IUL, Lisboa, 2016.
9. Что представляют собой данные лазерной съемки? [Электронный ресурс].— <http://desktop.arcgis.com/ru/arcmap/10.3/manage-data/las-dataset/what-is-lidar-data-.htm>. (Дата обращения: 17.03.2018)
10. S. Hrabar, ”Reactive obstacle avoidance for Rotorcraft UAVs IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011.
11. J. Saunders, B. Call, A. Curtis, R. Beard, ”Static and Dynamic Obstacle Avoidance in Miniature AirVehicles Brigham Young University, 2017

12. A. Barry and R. Tedrake, "Pushbroom Stereo for High-Speed Navigation in Cluttered Environments Robotics and Automation (ICRA) 2015

13. Устройство камеры видеонаблюдения, [Электронный ресурс].—
https://video-praktik.ru/kamery_ustrojstvo.html. (Дата обращения: 25.03.2018)

14. Эффект Доплера, [Электронный ресурс].—
https://ru.wikipedia.org/wiki/Эффект_Доплера (Дата обращения: 25.03.2018)