# 2 Название раздела

В данной главе производится описание разработанного алгоритма.

# 2.1 Описание входных данных алгоритма

В разработанном алгоритме используются множество датчиков определения расстояния. Данные датчики могут быть заменены, например, на датчики приближения.

На рисунке 1 представлен репрезентативный массив значений входных данных алгоритма, где по оси X расположены датчики с 1a по 1q, а по оси Y их показатели от 1 до \*.

Рисунок 1 - репрезентативный массив значений входных данных алгоритма

Примером датчика определения расстояния может служить VL53L0X, который работает на основе принципа лидара. Он использует инфракрасный лазер для измерения расстояния до объекта на основе времени, за которое отраженный лазерный сигнал возвращается обратно к датчику.

Набор сигналов от всех датчиков дает представление о контуре одного "среза" сканируемой области. Считывание нескольких срезов позволяет получить трехмерное представление о контуре сканируемой области.

# 2.2 Обзор алгоритма

Алгоритм предназначен для обработки данных от датчиков и обнаружения объектов в системе. Он включает в себя этапы фильтрации данных, контроля скорости прохождения объектов через датчики и обнаружения шарообразных объектов.

Первый этап - фильтрация данных. Все поступающие от датчиков показания проходят через специальный фильтр, который удаляет ненужные сигналы низкой интенсивности, шумы и помехи, оставляя только информацию о рассматриваемых объектах.

Второй этап - алгоритм скользящего окна с контролем скорости прохождения объектов через датчики. Датчики размещаются в определенном порядке и показания с них используются для определения скорости движения объектов в пределах системы. Для этого применяется скользящее окно, которое позволяет анализировать некоторый период времени и выяснить, сколько объектов проходит через датчики за данный промежуток времени. При этом высчитывается скорость прохождения.

Третий этап - алгоритм обнаружения объектов. После фильтрации данных и контроля скорости прохождения, происходит обнаружение объектов шарообразной формы.

# 2.3 Фильтрация данных

Для решения поставленной задачи фильтрации данных необходимо учесть ряд свойств и выбрать наиболее подходящий для данного случая фильтр. Анализ методов фильтрации, представленный выше, позволяет систематизировать подходы к обработке данных и выбрать оптимальный фильтр. Процесс фильтрации данных разделен на несколько этапов.

Первый этап – сбор данных. На данном этапе процесса фильтрации производится сбор и запись данных с датчиков для последующей обработки.

Далее следует этап предобработки данных. В процессе предобработки осуществляется фильтрация данных от возможных выбросов или шумов. Также учитываются физические ограничения, такие как предельные значения, и производится удаление данных с низкой интенсивностью. Этот этап позволяет подготовить данные к применению фильтра.

Последний этап – применение фильтра. Важным шагом в процессе фильтрации является выбор конкретного фильтра, который будет применяться к данным. Целью применения фильтра является удаление шумов, выбросов или несовершенств оборудования.

При выборе необходимого фильтра для поставленной задачи, необходимо учесть такие параметры, как частотный диапазон сигнала, требуемое подавление шума или выбросов, а также особенности конкретной системы или оборудования. Также следует учитывать вычислительные возможности доступных алгоритмов и время обработки данных.

Таким образом, для эффективной фильтрации данных с датчиков необходимо выбирать соответствующий фильтр, учитывая специфику сигнала и требования к итоговым данным. Подходящий выбор фильтра обеспечит эффективную фильтрацию шумов и артефактов, а также сохранение важных деталей сигнала, что в конечном итоге повысит качество и надежность результатов обработки сигналов.

Рисунок 2 демонстрирует важность порога вхождения (ПВ). Алгоритм требует установления ПВ, который фильтрует сигналы низкой интенсивности.

Рисунок 2 – Демонстрация важности ПВ

2b является показателем подходящего ПВ, поскольку он представляет собой уровень сигнала, превышающий меньшие части периметра яиц поскольку он не настолько высок, чтобы не указать на присутствие даже самого маленького яйца (или другого объекта), подлежащего сканированию. Теоретический ПВ на уровне, указанном 2c, был бы слишком низким, поскольку он был бы меньше, чем расстояние до самого маленького яйца. Поэтому это яйцо не будет подсчитано. Теоретический ПВ на уровне, обозначенном 2a будет слишком высоким, так как при таком уровне ПВ невозможно обнаружить "края" объектов.

# 2.4 Описание алгоритма обнаружения объектов

Основной принцип работы алгоритма обнаружения объектов заключается в детектировании и создании группы вокруг объекта, добавление датчиков в существующую группу, и поиск вершины путем выбора тройки датчиков с наименьшей суммой и определения центра этой тройки. Также алгоритм включает в себя отключение от группы на основе обнаружения другого объекта, корректировку принадлежности датчиков к группам путем проверки экстремумов, и отключение датчиков при прохождении объектов.

Блок-схема алгоритма представлена на рисунке 3.

Рисунок 3 – Блок-схема алгоритма

Ниже приводится краткое пошаговое описание алгоритма в отношении показаний матрицы сигналов, показанных на рисунке 4.

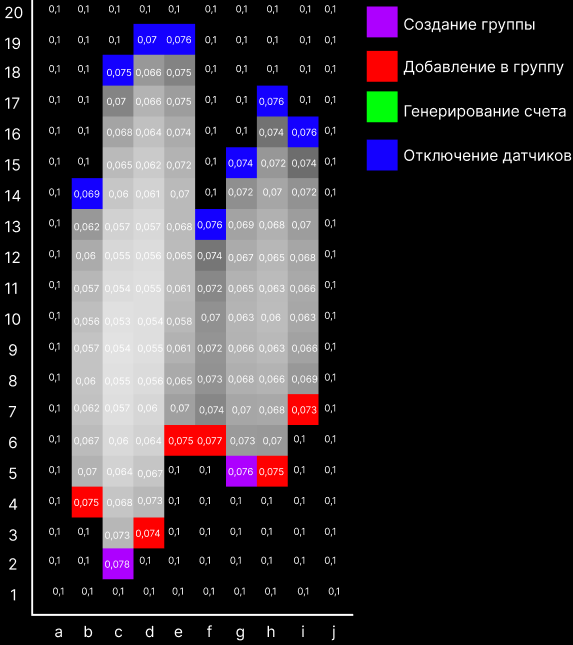


Рисунок 4 - Показания матрицы сигналов.

Показание 1. Ни одно показание не превысило порог вхождения, все датчики остаются выключенными.

Показание 2. Показание датчика c превышает порог вхождения. Так как по соседству отсутствуют созданные группы, то создается новая группа (группа №1). Центр данной группы устанавливается на датчик c.

Показание 3. Показание датчика d превышает порог вхождения. Так как по соседству существует группа, то датчик d добавляется в данную группу (группа №1). Центр данной группы устанавливается на датчик c.

Показание 4. Показание датчика b превышает порог вхождения. Так как по соседству существует группа, то датчик b добавляется в данную группу (группа №1). Центр данной группы устанавливается на датчик c.

Показание 5. Показание датчика g превышает порог вхождения. Так как по соседству отсутствуют созданные группы, то создается новая группа (группа №2). Показание датчика h превышает порог вхождения. Так как по соседству существует группа, то датчик h добавляется в данную группу (группа №2). Центр группы 1 устанавливается на датчик с, а центр группы 2 устанавливается на на датчик h.

Показание 6. Показание датчика e превышает порог вхождения. Так как по соседству существует группа, то датчик e добавляется в группу №1. Показание датчика f превышает порог вхождения. Так как рядом находятся 2 группы, то датчик присоединятся к той, где показатель соседнего датчика меньше. Центр группы 1 устанавливается на датчик с, а центр группы 2 устанавливается на датчик h.

Показание 7. Показание датчика i превышает порог вхождения. Так как рядом находится группа №2, то датчик присоединяется к ней. Центр группы 1 устанавливается на датчик с, а центр группы 2 устанавливается на на датчик h.

Показание 8-11. Центр группы 1 устанавливается на датчик с, а центр группы 2 устанавливается на на датчик h.

Показание 12. Значение тройки датчиков с наименьшей суммой группы 2 упало больше, чем на 5%. Это означает, что найдена вершина объекта и генерируется счет. Всем датчикам в этой группе устанавливается флаг, который запрещает генерировать счет. Центр группы 1 устанавливается на датчик с.

Показание 13. Показание датчика f упало ниже порога вхождения. Данный датчик отключается. Центр группы 1 устанавливается на датчик с.

Показание 14. Значение тройки датчиков с наименьшей суммой группы 2 упало больше, чем на 5%. Это означает, что найдена вершина объекта и генерируется счет. Всем датчикам в этой группе устанавливается флаг, который запрещает генерировать счет. Показание датчика b упало ниже порога вхождения. Данный датчик отключается.

Показание 15. Показание датчика g упало ниже порога вхождения. Данный датчик отключается.

Показание 16. Показание датчика i упало ниже порога вхождения. Данный датчик отключается.

Показание 17. Показание датчика h упало ниже порога вхождения. Данный датчик отключается.

Показание 18. Показание датчика c упало ниже порога вхождения. Данный датчик отключается.

Показание 19. Показание датчиков d, e упали ниже порога вхождения. Данные датчики отключаются.

# 2.5 Описание алгоритма скользящего окна с контролем скорости

При наличии шума или непостоянной скорости движения объектов, стандартные методы фильтрации могут давать неточные результаты.

Один из аналогов алгоритма скользящего окна с контролем может быть применен для решения подобных проблем. Этот алгоритм, основываясь на предположении о непрерывности скорости движения объектов, позволяет более точно определить их скорость и направление движения для корректной работы алгоритма обнаружения объектов.

Принцип работы алгоритма состоит в том, что он использует окно заданного размера, которое скользит по временному ряду данных. В каждой позиции окна производится анализ данных на предмет наличия объектов, и на основании этого производится оценка их скорости. Размер окна зависит от объектов и диапазона скорости. Окно должно быть достаточным размером, чтобы про изменении скорости алгоритм успел рассчитать скорость.

Основное преимущество данного алгоритма состоит в том, что он учитывает изменение скорости движения объектов во времени и способен дать более точные результаты при наличии шума или непостоянной скорости. Схема работы изображен на рисунке 5.

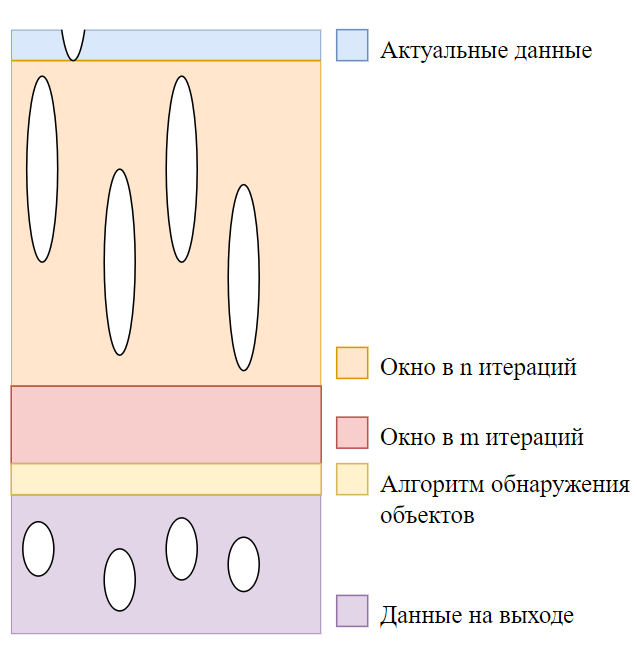


Рисунок 5 – Схема работы алгоритма классификации и скользящего окна

Блок-схема алгоритма представлена на рисунке 6.

Рисунок 6 – Блок-схема алгоритма

Ниже приводится краткое пошаговое описание алгоритма классификации в отношении показаний матрицы сигналов, показанных на рисунке N.