В разработанном алгоритме используются множество датчиков расстояния. Каждый датчик определяет расстояние на основе различных технологий, таких как ультразвук, инфракрасный лазер, оптическое или радио частотное излучение. Например, датчик определения расстояния VL53L0X 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.

Когда лазерный сигнал выходит из датчика и попадает на объект, он отражается от него и возвращается обратно к датчику. Датчик затем измеряет время, за которое произошел этот процесс, и использует его для вычисления расстояния до объекта. Метод измерения расстояния довольно точен и позволяет датчику работать на расстоянии до нескольких метров.

Датчик VL53L0X также обладает функцией автоматической компенсации изменений окружающих условий, таких как освещенность и температура, что помогает обеспечить стабильную и точную работу в различных условиях.

Набор сигналов от всех датчиков дает представление о контуре одного "среза" сканируемой области. Считывание нескольких срезов позволяет получить трехмерное представление о контуре сканируемой области. Алгоритм может быть использован для подсчета количества объектов в сканируемой области, а также для вычисления объема объектов. Когда изобретение используется для подсчета объектов на поверхности, каждый объект должен быть окружен областью, которая отражает меньше света, чем сами объекты. Объекты могут касаться друг друга, если область, где они соприкасаются, отражает меньше света, чем остальные объекты.

Данные от всех датчиков объединяются для создания изображения среза сканируемой области. Путем считывания нескольких срезов можно получить трехмерное представление контура сканируемой области. Это устройство может использоваться для подсчета числа объектов в сканируемой области и определения их объема. При подсчете объектов, находящихся на поверхности, каждый объект должен быть окружен областью с более низким уровнем отражения света, чем сами объекты. Объекты могут соприкасаться друг с другом, если площадь контакта отражает меньше света, чем остальные объекты.

В качестве альтернативы датчиков расстояния могут использоваться датчики интенсивности отраженного света, которые измеряют уровень отраженного света от объекта и могут использоваться для определения расстояния до него.

Алгоритм идеально подходит для подсчета и определения размеров объектов, движущихся по движущейся конвейерной ленте, без необходимости размещения объектов в один ряд.

Рисунок N представляет собой вид в вертикальном разрезе платы с датчиками расположенного над лентой для переноса объектов.

Датчики с Na по Nh расположены на одинаковом расстоянии и установлены на плате N1. Плата N1 расположена над лентой N2.

Объекты N3 и N4 перемещаются по ленте N2 мимо датчиков с Na по Nh. Такие объекты не обязательно должны быть расположены в один ряд для того, чтобы измерения могли быть выполнены в соответствии с алгоритмом. Каждый датчик опрашивается и фиксирует показатели перпендикулярно плоскости датчика и в направлении ленты N2.

Каждый цикл опроса всех датчиков приводит к считыванию «среза области» под датчиками в определенное время. По мере перемещения объектов вдоль ленты N2 могут быть получены дополнительные "срезы" показаний. Таким образом, каждый датчик может рассматриваться как представитель координаты X, каждое последовательное считывание может рассматриваться как представитель координаты Y ля конкретной координаты X, а фактический сигнал датчика может рассматриваться как представитель координаты Z сканируемой области.

Рисунок N2 представляет собой график, показывающий показания для одного опроса каждого датчика ленты для переноса объектов, показанной на рисунке N. По оси X представлены датчики с N2a по N2h, а по оси Y – расстояния от датчиков до объектов.

Количество датчиков, выбранных для прохождения траектории для конкретного применения, зависит от нескольких характеристик, таких как ширина траектории, размер объектов, количество объектов, которые могут поместиться на траектории, и желаемое разрешение. Как правило, большее разрешение предпочтительно для вычисления объемов объектов, в то время как меньшее разрешение необходимо только для подсчета объектов.

Расстояние между каждым датчиком также важно, когда изобретение используется для подсчета объектов. Изобретение подсчитывает объект путем обнаружения смежной области. На рисунке N изображен контурный чертеж ленты из

-зависит от шума

-от размера объектов

-от технической спецификации

ОПИСАТЬ РАССТОЯНИЕ РАСПОЛОЖЕНИЯ ДАТЧИКОВ

Рисунок N2 также демонстрирует важность порога вхождения ОПИСАТЬ ВАЖНОСТЬ ПВ

На рисунке N3 представляет собой вид сверху типичного пути, содержащего объекты. На этом рисунке яйца 60, 61 и 62 лежат на ленте 63, которая движется мимо сканера (не показан). Яйца находятся выше (ближе к сканеру) в своих центральных частях, чем в периферийных частях. Яйца генерируют показания, показанные на рисунке N4.

На рисунке N4 представлен репрезентативный массив значений сигналов, указывающих на обнаруженный контур траектории, показанной на рисунке N3. Обнаружение объектов осуществляется путем обнаружения перепадов в отдельных датчиках и группировки датчиков, которые также демонстрируют изменения в показаниях.

Основной принцип работы алгоритма заключается в обнаружении 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 n.

Рисунок n – Блок-схема алгоритма

Изначально все датчики выключены, что означает, что они не могут генерировать счет. Каждый датчик имеет отдельные поля.

* value – текущее значение;
* deltaY – описание;
* center – логическое значение, которое определяет является ли данный датчик центром тройки;
* listhen – логическое значение, которое отображает включен ли датчик;
* isFindTop – логическое выражение, которое определяет найдена ли вершина;
* isEggOne - – логическое выражение, которое определяет был ли у датчиков переход на другой объект;
* iterEggCol – количество показаний данной группы;
* idGroup – отображает номер группы;

Ниже приводится краткое пошаговое описание алгоритма в отношении показаний карты вершин, показанных на рисунке N.

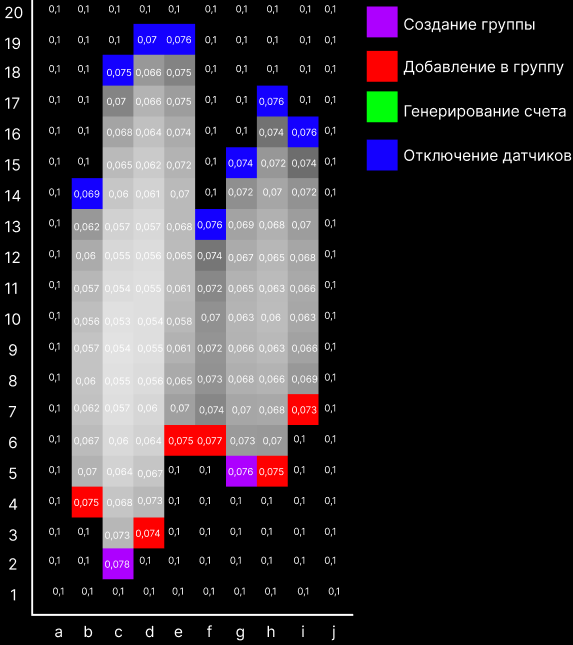


Рисунок N - Показания карты вершин.

Рисунок N представляет собой репрезентативный массив значений сигналов.

ПОШАГОВОЕ ОПИСАНИЕ КАК В ПАТЕНТЕ

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 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.