

**Общее описание
Программы для ЭВМ «Программа для измерения насыпного
объема сыпучих материалов на подвижных или
статистических поверхностях»**

Оглавление

Термины и определения	3
1. Введение	4
2. Назначение Программы.....	5
3. Ключевые функции.....	6
4. Описание интерфейса администрирования Программы	7
4.1. Авторизация подключения	7
4.1.1. Авторизации подключения к основному ПО.....	7
4.1.2. Авторизации подключения к модулю аппаратного контроля зависания ПО и вычислительного модуля	8
4.2. Первичная настройка нулевого объема	8
4.2.1. Для запуска скриптов:	8
4.2.2. Первичная настройка нулевого объема (5 шагов).....	8
4.3. Основной режим работы по видеофиксации и расчету насыпного объема	16
4.4. Контроль работоспособности и доступности ПО	17
4.5. Контроль работоспособности и доступности вычислительного модуля	18

Термины и определения

Термин	Определение
CV	Computer Vision
АСУ ТП	Автоматическая система управления технологическим процессом
БД	База данных
ПК	Персональный компьютер
ПО	Программное обеспечение
Программа	Программа для измерения насыпного объема сыпучих материалов на подвижных или статистических поверхностях
СОД	Сервер обработки данных
ОС	Операционная система
СУБД	Система управления базами данных
Пайплайн	Последовательность действий и процессов, выполняемых при запуске программы, для достижения поставленной задачи.
SSH	Secure Shell - сетевой протокол прикладного уровня, предназначенный для безопасного удаленного доступа
VNC	Virtual Network Computing - метод удаленного доступа к рабочему столу компьютера по сети

1. Введение

Настоящий документ является описанием Программы для ЭВМ «Программы для измерения насыпного объема сыпучих материалов на подвижных или статистических поверхностях» (далее – «Программа», «ПО»).

Программа предназначена для измерения насыпного объема сыпучих материалов на подвижных или статистических поверхностях с помощью видеофиксации данных и последующей обработки видеопотока для расчета объема.

Рассчитанные в Программе объемы передаются в формате, необходимом для интеграции с автоматизированной системой управления технологическими процессами (АСУТП).

Программа разворачивается на базе операционной системы Raspberry Pi OS 64-bit, в качестве локальной СУБД используется MariaDB, удаленной - PostgreSQL.

Одним из основных требований для обеспечения работы Программы является наличие внешнего статического IP-адреса (возможны варианты с перенаправлением потоков на внутренний адрес через прокси-сервер).

2. Назначение Программы

Программа предназначена для измерения объемов сыпучих материалов на статистических и подвижных агрегатах технологического процесса, например, на конвейерной ленте, грохоте и т.д. с использованием камеры глубины и вычислительного модуля для обработки видеопотока с камеры. Рассчитанные в Программе объемы предоставляются по сети Ethernet для возможности интеграции данных с автоматизированной системой управления технологическими процессами (АСУТП).

Логика работы Программы заключается в первоначальной установке нулевого объема, характеризующего геометрию измеряемой поверхности агрегата, которая производится при первичной настройке Программы. Далее Программа измеряет объем проходящего через область съемки (замера) сыпучего материала, и вычитает из этого объема нулевой объем из настроек. Такой расчет дает фактическое значение объема проходящего сыпучего материала.

Программа может использоваться самостоятельно для целей сбора и мониторинга информации от подключенных объектов, либо в составе комплекса программных продуктов путем интеграции с ними для выработки рекомендательных действий оператору технологического процесса.

3. Ключевые функции

Основные функции Программы:

- Видеофиксация сырья в потоке;
- Контроль остановки движения потока сырья;
- Сбор, хранение и обработка, с учетом остановки движения сырья, видеоданных, поступающих с камеры глубины, установленной в точке контроля технологического процесса;
- Расчет насыпного объема сырья в потоке;
- Передача актуальных данных текущего состояния технологического процесса в АСУ ТП посредством Ethernet/IP протокола;
- Контроль зависания ПО и вычислительного модуля.

4. Описание интерфейса администрирования Программы

Работа в интерфейсе администрирования предполагает решение следующих задач:

- Первичная настройка нулевого объема;
- Основной режим работы по видеофиксации и расчету насыпного объема;
- Программный и аппаратный контроль зависания ПО и вычислительного модуля.

4.1. Авторизация подключения

4.1.1. Авторизации подключения к основному ПО

Авторизация и подключение к основному ПО осуществляется с помощью протокола SSH или метода удаленного доступа VNC. В обоих случаях необходимо использовать следующие данные:

- IP адрес устройства (например, 10.95.105.103);
- Логин (например, forkuser);
- Пароль (например, admin).

Для подключения с помощью SSH используется команда следующего вида, которую следует ввести в окне терминала:

ssh username@ip -p port ,

где *username* – имя пользователя, *ip* – ip-адрес устройства, *port* – порт.

После чего в открывшемся окне появится строка для ввода пароля. Для подключения с помощью VNC необходимо скачать программу RealVNC Viewer. Потребуется ввести IP-адрес, после чего в открывшемся окне ввести имя пользователя и пароль.

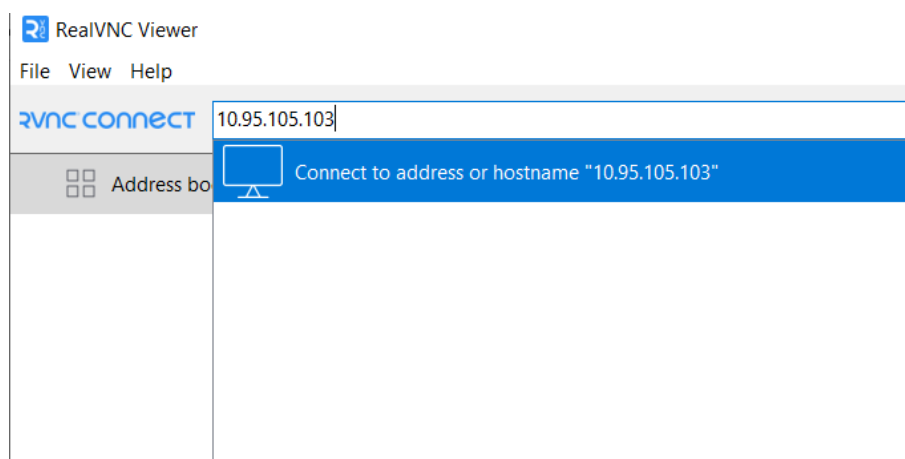


Рис.1. Ввод IP-адреса в программе RealVNC Viewer.

4.1.2. Авторизации подключения к модулю аппаратного контроля зависания ПО и вычислительного модуля

Авторизация и подключение к модулю аппаратного контроля зависания ПО и вычислительного модуля осуществляется с помощью Web-интерфейса через браузер. Необходимо использовать следующие данные:

- IP адрес устройства аппаратного контроля (например, 10.95.105.103);
- Логин (например, forkuser);
- Пароль (например, admin).

4.2. Первичная настройка нулевого объема

Для запуска алгоритма расчета объема требуется выполнить 6 шагов, для каждого из которых были написаны соответствующие python-скрипты. В связи с тем, что при включении прибора, автоматически выполняется запуск пайплайна для расчета объема, необходимо остановить этот процесс и освободить подключение к камере глубины, выполнив команду в окне терминала:

```
sudo bash /home/forkmeter/Projects/scripts/kill_pipe.sh
```

4.2.1. Для запуска скриптов:

а. Необходимо зайти на устройство через VNC Viewer. Для этого потребуется ввести ip-адрес, имя пользователя и пароль.

б. Запуск скриптов осуществляется с помощью виртуального окружения python версии 3.9. Для этого открываем терминал и переходим в папку проекта:

```
cd /home/forkmeter/Projects, здесь forkmeter - имя пользователя.
```

с. Активировать виртуальное окружение:

```
source venv/bin/activate
```

д. Далее можно приступить к настройке Программы.

4.2.2. Первичная настройка нулевого объема (5 шагов)

4.2.2.1. Запись bag-файла при движущемся сырье в кадре:

```
python3.9 record_bag.py 1
```

После запуска скрипта открывается окно с предварительным просмотром кадра, нужно нажать любую клавишу, чтобы начать запись.



Рис.2. RGB- изображение конвейера с углем.

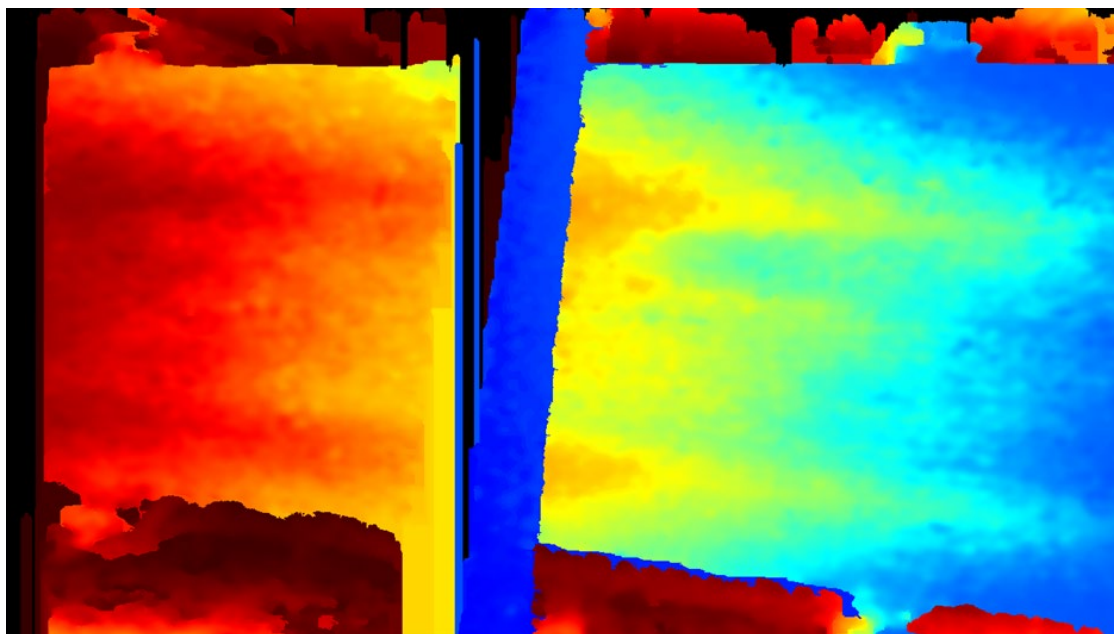


Рис.3. Изображение карты глубины на конвейере с углем.

Запись сохраняется в папку: `data/init_bags`.

4.2.2.2. Запись bag-файла при пустом конвейере или грохоте.

```
python3.9 record_bag.py 0
```

После запуска скрипта открывается окно с предварительным просмотром кадра, нужно нажать любую клавишу, чтобы начать запись.

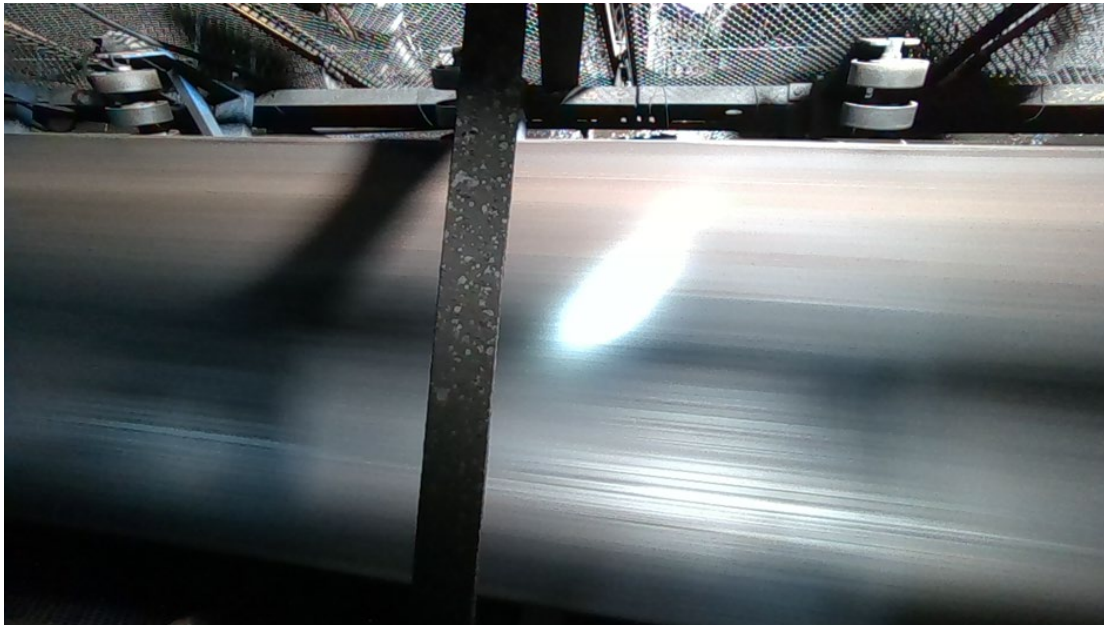


Рис.4. RGB- изображение пустого конвейера.

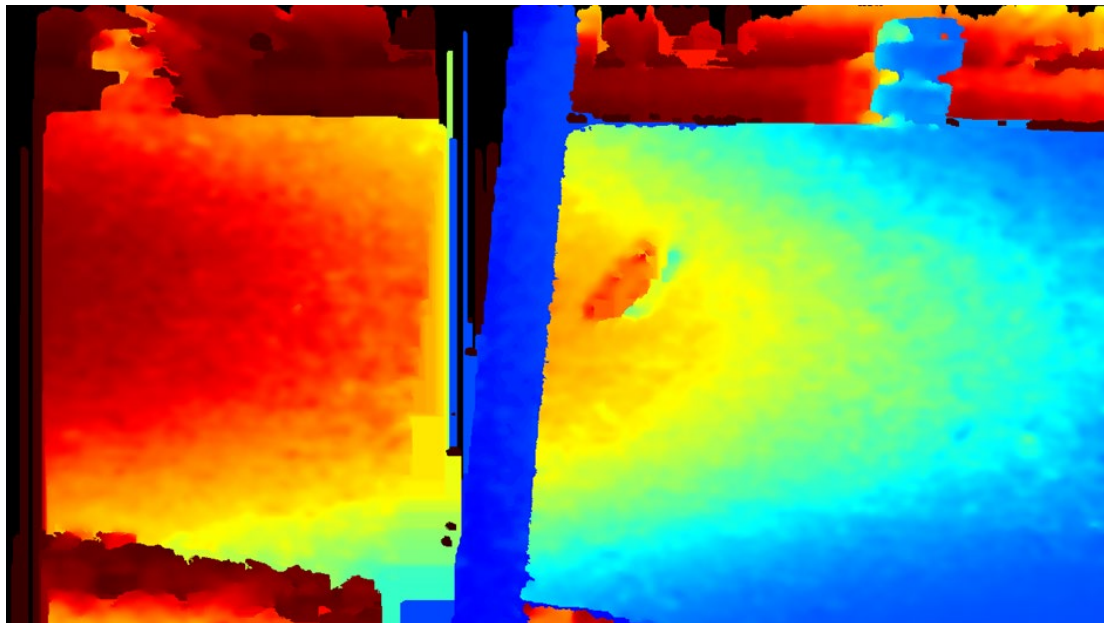


Рис.5. Изображение карты глубины пустого конвейера

Запись сохраняется в папку: *data/empty_bags*.

4.2.2.3. Определение скорости перемещения сырья в кадре.

Данное значение описывает сколько пикселей проходит сырье за 1 кадр. По умолчанию расчеты выполняются для скорости съемки 5 кадров в секунду.

Есть 2 варианта расчета скорости перемещения сырья, в зависимости от места съемки: на грохоте или на конвейере:

а. Настройка скорости перемещения для грохота

python3.9 setup_flow.py

Выполняется автоматический расчет попиксельного смещения угла в кадре и открывается окно с просмотром изображения карты глубины.

Необходимо зажать левую кнопку мыши и белым цветом провести линию, параллельную интересующей нас области. Это необходимо для расчета угла поворота кадра. Далее последовательно необходимо нажать *клавиши w и d*.

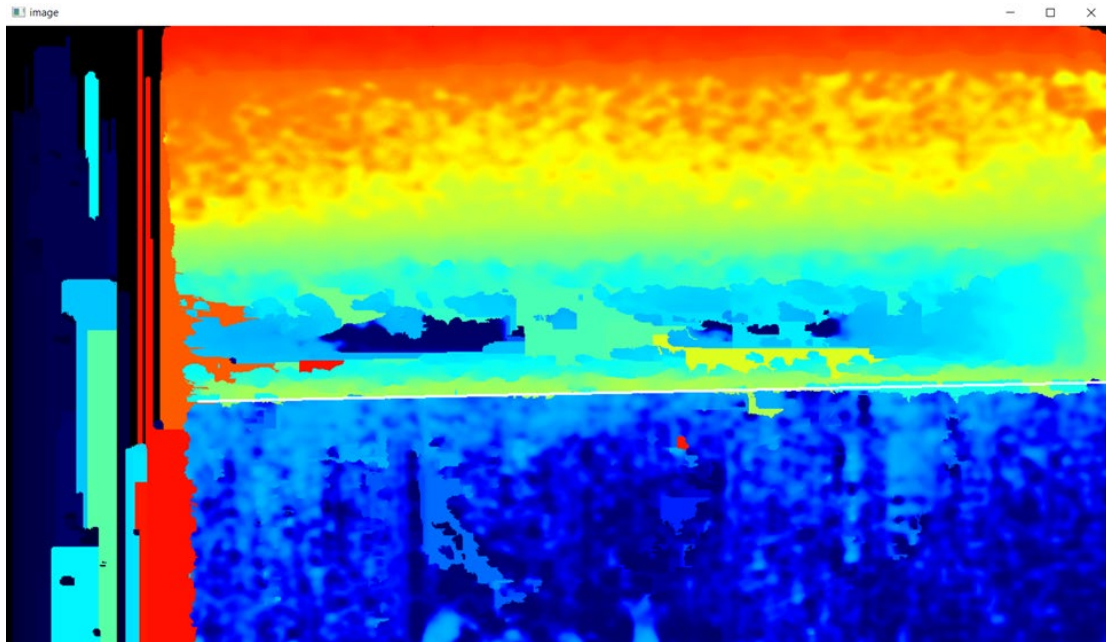


Рис.6. Определение угла поворота на изображении карты глубины

В следующем открывшемся окне белым прямоугольником необходимо обозначить рабочую область, которая будет использоваться при расчете объема. После этого нажимается *клавиша w* и окно закрывается.

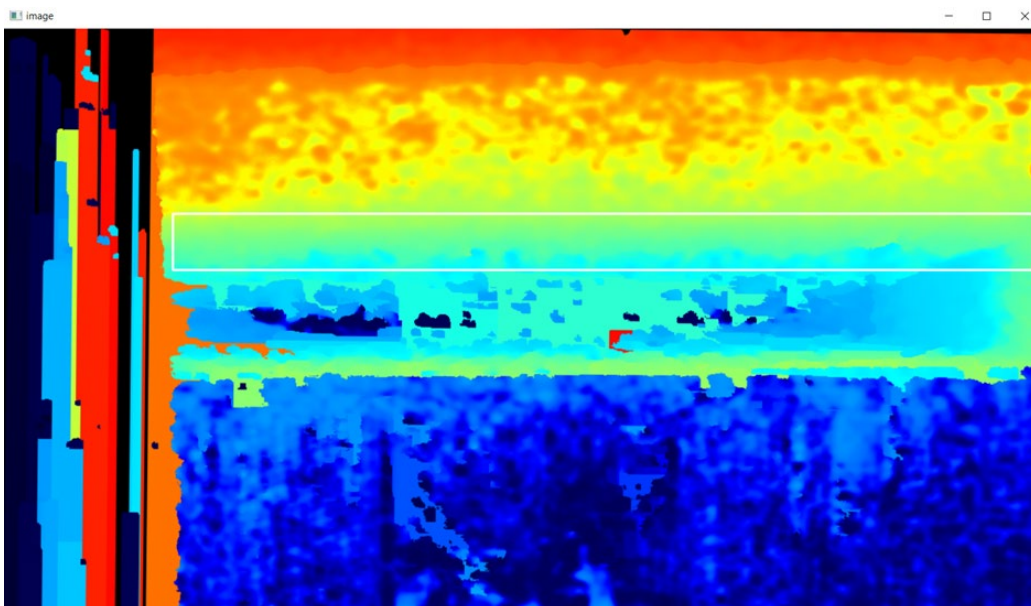


Рис.7. Ограничение рабочей области прямоугольником

в. Настройка скорости для конвейера

python3.9 setup_move.py

Выполняется выделение углей прямоугольниками и расчет попиксельного смещения за 1 кадр.

Белой линией определяется угол поворота изображения. Необходимо последовательно нажать *клавиши w и d*.

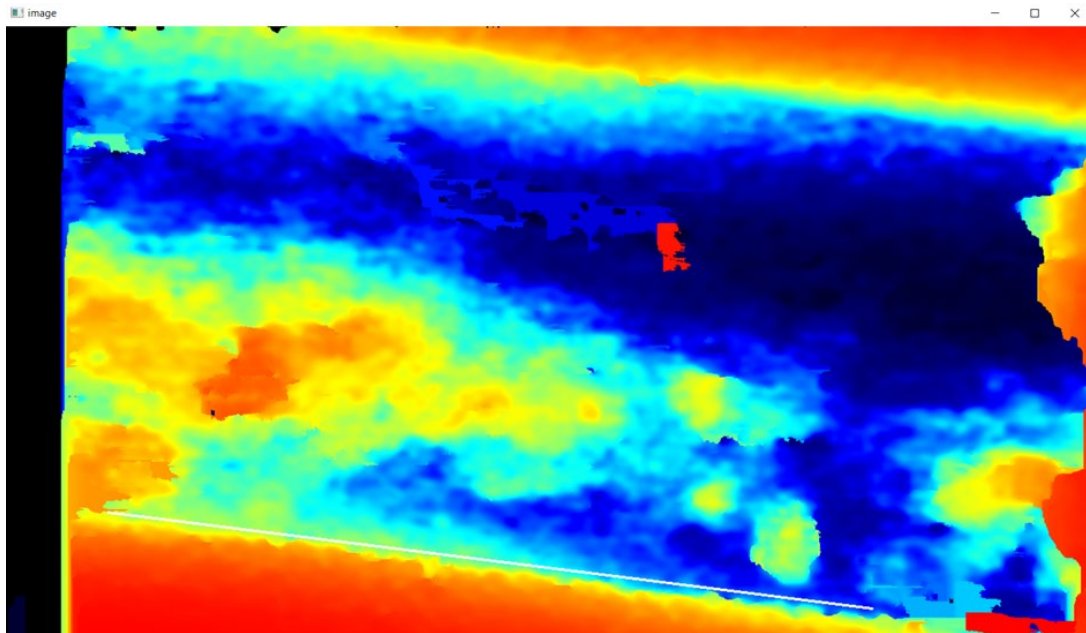


Рис.8. Определение угла поворота на изображении карты глубины

В открывшемся окне выделяется прямоугольником крупный экземпляр угля и нажимается *клавиша w*.

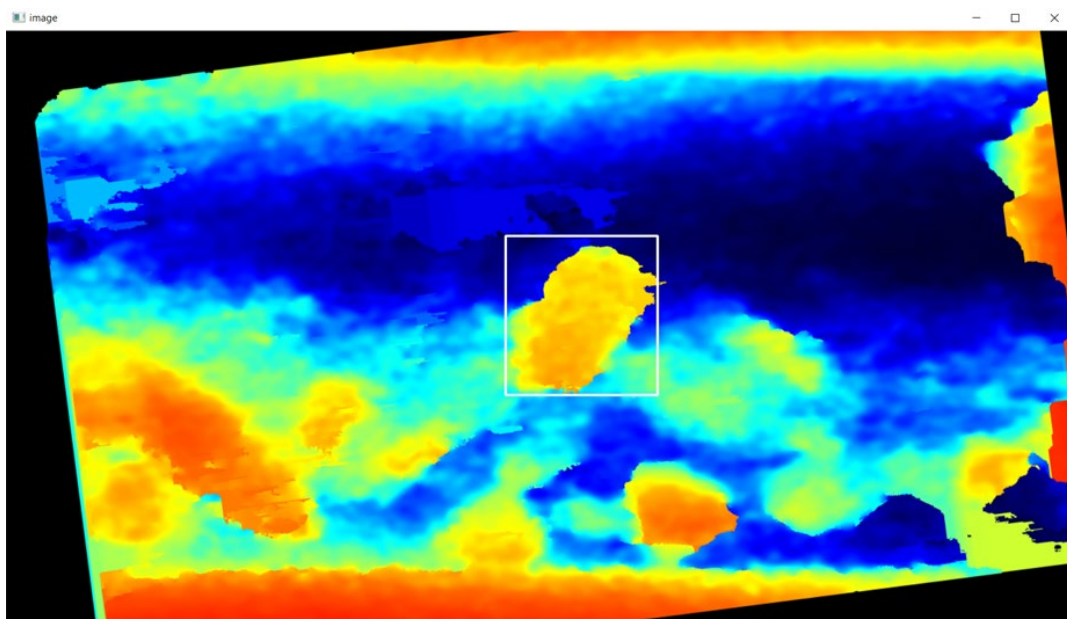


Рис.9. Определение угла поворота на изображении карты глубины

Далее необходимо повторить действие на следующем кадре. Выделяется тот же экземпляр угля, что и на предыдущем шаге, затем последовательно нажимаются *клавиши w и d*.

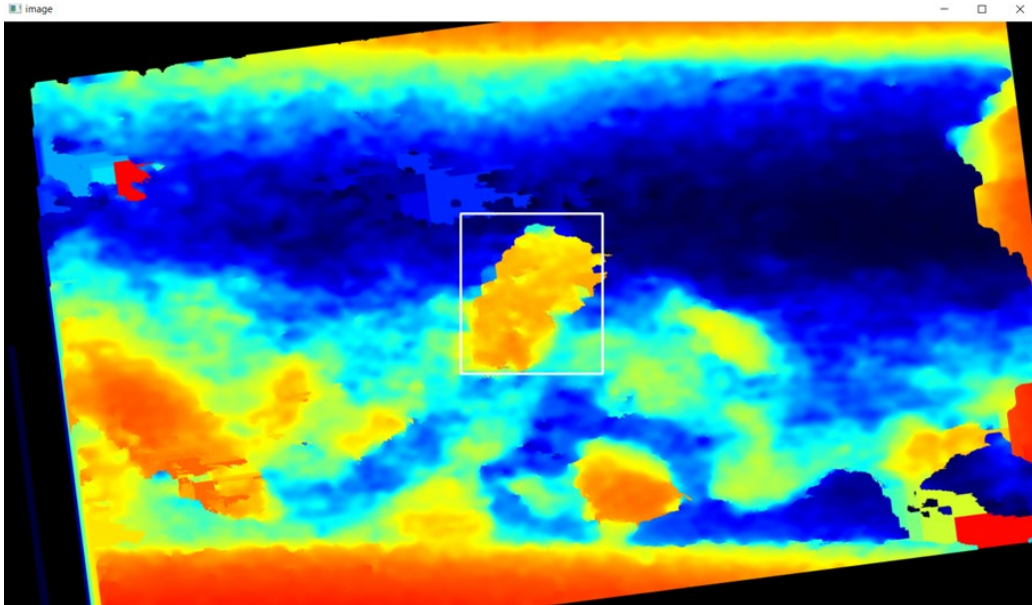


Рис.10. Определение угла поворота на изображении карты глубины

4.2.2.4. Определение рабочей области и расчет значений объема пустой области под камерой - установка нулевого объема.

`python3.9 setup_empty.py`

Открывается окно с просмотром изображения карты глубины. Необходимо зажать левую кнопку мыши и белым цветом провести линию, параллельную интересующей нас области — это необходимо для расчета угла поворота кадра. Далее необходимо последовательно нажать *клавиши w и d*.

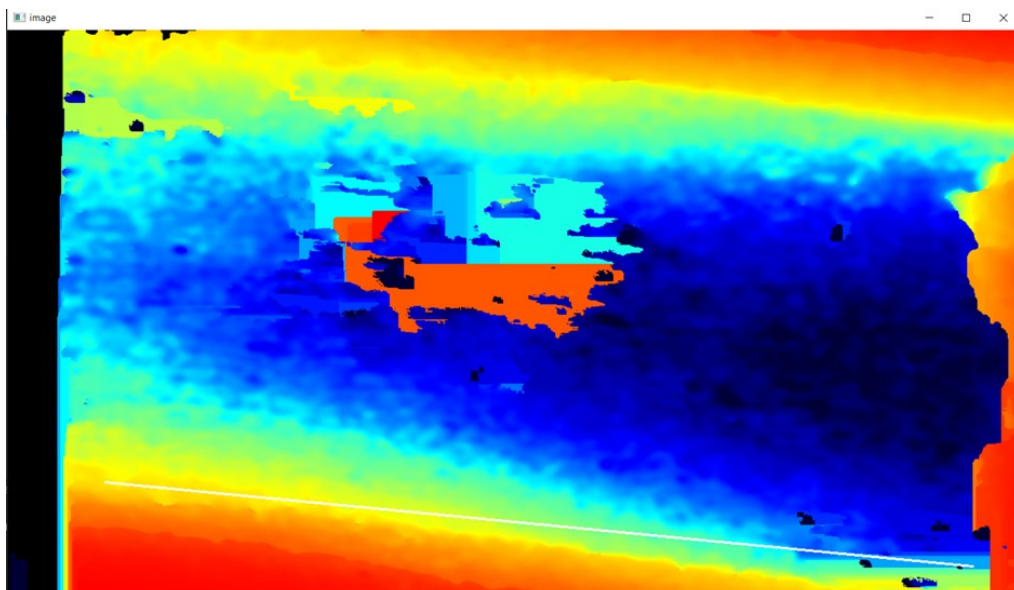


Рис.11. Определение угла поворота на изображении карты глубины

В следующем открывшемся окне белым прямоугольником необходимо обозначить рабочую область, которая будет использоваться при расчете объема. После этого нажимается *клавиша и* и окно закрывается.

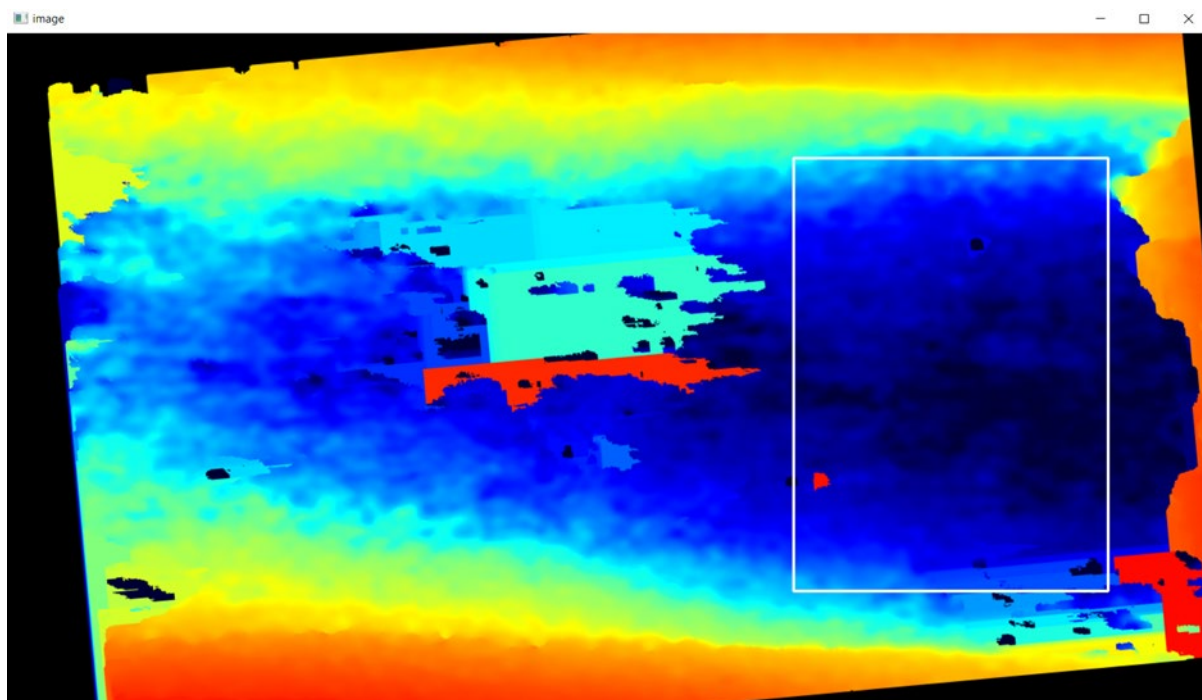


Рис.12. Ограничение рабочей области прямоугольником

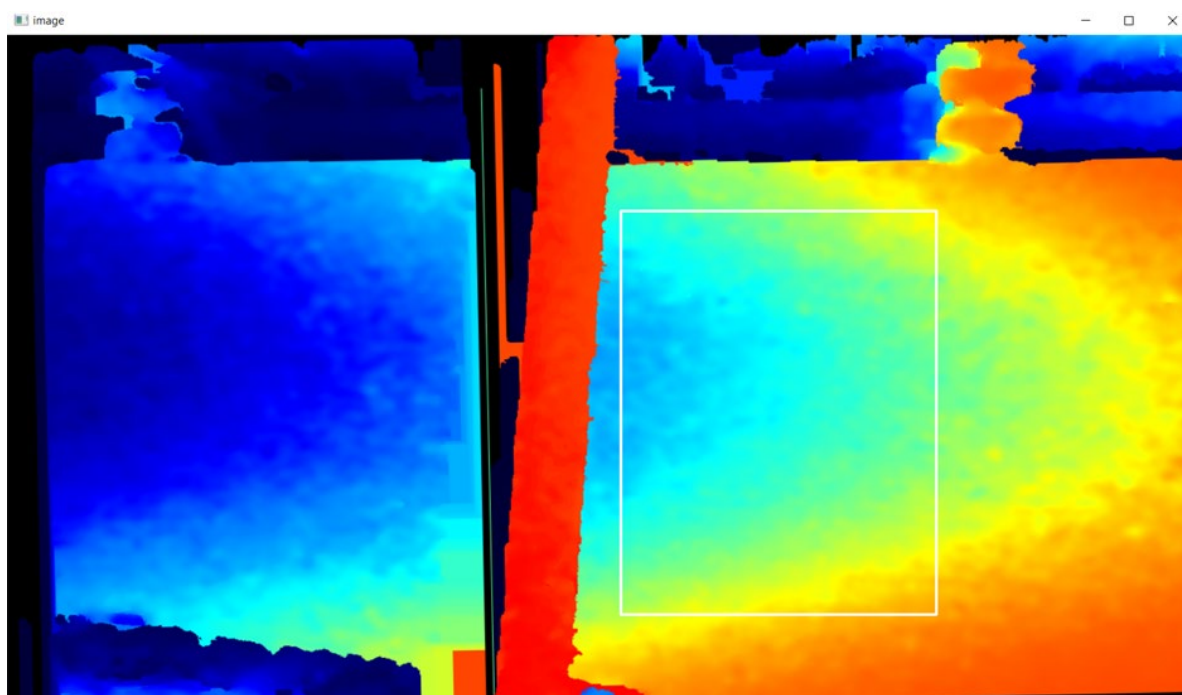


Рис.13. Ограничение рабочей области прямоугольником

В результате выполнения шагов 3 и 4 рассчитанные значения сохраняются в файл *configs/setup.txt*.

```
y1=557
mean_center_depth=1302.0
median_center_depth=1301.0
SCALE_COEFF_X=0.000973333
SCALE_COEFF_Y=0.001027391
volume0=0.191188133
rotate_angle=-13.0
volume0_raw=191189.36
volume0_window=0.191540178

x0=657
y0=169
x1=1048
y1=582
mean_center_depth=1283.0
median_center_depth=1285.0
SCALE_COEFF_X=0.000960766
SCALE_COEFF_Y=0.001013888
volume0=0.201888264
rotate_angle=-5.7
volume0_raw=207254.26
volume0_window=0.201844511

fps=5
px_velocity_x=892
px_velocity_y=5
x0=168
y0=129
x1=702
y1=485
mean_center_depth=1392.0
median_center_depth=1389.0
SCALE_COEFF_X=0.001039863
SCALE_COEFF_Y=0.001098878
volume0=0.302589398
rotate_angle=0.0
volume0_raw=264806.17
volume0_window=0.305129321
```

Рис.14. Файл *setup.txt*

Итоговые значения с конфигурацией необходимо скопировать и вставить в файл *configs/setup.ini*.

```
[cfg]
fps=5
px_velocity_x=500
px_velocity_y=200
x0=657
y0=169
x1=1048
y1=582
mean_center_depth=1283.0
median_center_depth=1285.0
SCALE_COEFF_X=0.000960766
SCALE_COEFF_Y=0.001013888
volume0=0.201888264
rotate_angle=-5.7
volume0_raw=207254.26
volume0_window=0.201844511
```

Рис.15. Пример итогового конфиг-файла *setup.ini*

4.2.2.5. Настройка БД

Финальная настройка обработки и записи в БД осуществляется в конфигурационном файле *configs/fork_cfg.ini* - там необходимо указать

порядковый номер экземпляра форкметра (fork_id) и ориентацию камеры относительно рабочей поверхности: horizontal либо vertical. При установке камеры широкой частью параллельно движению конвейерной ленты - horizontal, при установке камеры широкой частью перпендикулярно движению угля на грохоте - vertical.

Далее осуществляется настройка локальной и удаленной БД. В качестве локальной БД используется MariaDB, удаленная - PostgreSQL.

```
[basic]
name = forkmeter
camera_position = horizontal
fork_id = 1

[db_local]
host=
user=
password=
db=results
charset=utf8mb4
cursorclass=pymysql.cursors.DictCursor

[db_serv]
database=results
user=
password=
host=
port=
```

Рис.16. Пример настройки БД

4.3. Основной режим работы по видеофиксации и расчету насыпного объема

Основной режим работы запускается с помощью скрипта run_pipeline_db.py. Для запуска программы необходимо в открытом окне терминала ввести команду:

```
python3.9 run_pipeline_db.py
```

Данный скрипт устанавливает подключение к камере и запускает обработку последовательности карт глубины для последующего расчета объема. Выполняется подключение и отправка результатов в локальную БД и в БД на сервере. Полученные мгновенные значения объемов записываются в поле “vol_m3”, время указывается в поле “time”, среднее значение глубины в кадре характеризует среднее расстояние до объекта в зоне видимости и хранится в поле

“mean_depth”, также для последующей обработки результатов записывается порядковый номер устройства – “fork_id”.

Свойства Данные Диаграмма						
volumes Введите SQL выражение чтобы отфильтровать результаты						
Таблица	123 id ↓	time	123 vol_m3	123 mean_depth	123 fork_id	
545	4 133 092	2024-09-11 19:47:29.681 +0300	0,20961343	525	4	
546	4 133 091	2024-09-11 19:47:25.074 +0300	0,19786799	566	4	
547	4 133 090	2024-09-11 19:47:20.314 +0300	0,084926166	630	4	
548	4 133 089	2024-09-11 19:47:15.280 +0300	0,19752166	564	4	
549	4 133 088	2024-09-11 19:47:10.546 +0300	0,20968634	526	4	
550	4 133 087	2024-09-11 19:47:05.896 +0300	0,14741823	603	4	
551	4 133 086	2024-09-11 19:47:00.695 +0300	0,19720909	559	4	
552	4 133 085	2024-09-11 19:46:56.157 +0300	0,2004451	530	4	
553	4 133 084	2024-09-11 19:46:51.329 +0300	0,21029773	529	4	
554	4 133 083	2024-09-11 19:46:46.721 +0300	0,21005	520	4	
555	4 133 082	2024-09-11 19:46:42.183 +0300	0,20647731	540	4	

Рис.17. Пример таблицы с результатами вычисления объемов в БД

4.4. Контроль работоспособности и доступности ПО

Для контроля перегрева устройства используется программный код, отслеживающий температуру процессора микрокомпьютера каждые 10 секунд. Критическим значением температуры процессора считается 90 °С. При нагреве до температуры 80% от критической, частота вычислений снижается в 2 раза до тех пор, пока температура устройства не примет допустимые значения. В случае повышения температуры до 90% от критического значения, вычисления приостанавливаются до тех пор, пока устройство не охладится до 80% от критического значения.

Свойства Данные Диаграмма					
temperature Введите SQL выражение чтобы отфильтровать результаты					
Таблица	123 id ↓	time	123 temp_cpu	123 temp_sensor	123 fork_id
1	394 939	2024-09-11 19:52:28.000 +0300	37	28	4
2	394 938	2024-09-11 19:52:16.000 +0300	37,4	28	4
3	394 937	2024-09-11 19:52:04.000 +0300	37,4	28	4
4	394 936	2024-09-11 19:51:51.000 +0300	37	28	4
5	394 935	2024-09-11 19:51:39.000 +0300	37,4	28	4
6	394 934	2024-09-11 19:51:27.000 +0300	36,5	28	4
7	394 933	2024-09-11 19:51:15.000 +0300	37,4	28	4
8	394 932	2024-09-11 19:51:02.000 +0300	37	28	4
9	394 931	2024-09-11 19:50:50.000 +0300	37	28	4
10	394 930	2024-09-11 19:50:38.000 +0300	38,4	28	4
11	394 929	2024-09-11 19:50:26.000 +0300	38,4	28	4
12	394 928	2024-09-11 19:50:14.000 +0300	37,9	28	4
13	394 927	2024-09-11 19:50:01.000 +0300	37,4	28	4

Рис.18. Пример таблицы в БД со значениями температуры на процессоре и внутри корпуса

Контроль работы устройства выполняется путем проверки поступления записей в локальную БД с результатами измерения объемов. В случае обнаружения отсутствия новых записей на протяжении более 3-х минут устройство перезагружается.

Отслеживание температуры и контроль поступления записей в БД осуществляется с помощью запуска python-скрипта *monitor_db.py* в терминале:

```
python3.9 monitor_db.py
```

4.5. Контроль работоспособности и доступности вычислительного модуля

Для контроля зависания ПО и вычислительного модуля также используется отдельный программный модуль *forkmeter.yaml*, исполняемый на дополнительном вычислительном устройстве (аналог Arduino) и отслеживающий подачу пульсирующего сигнала (логической единицы) на один из выходов GPIO основного вычислительного модуля. В случае прерывания пульсирующего сигнала на контрольном выводе программный модуль контроля зависания ПО и вычислительного модуля выдает команду на кратковременное отключение подачи питания на основной вычислительный модуль, инициируя таким образом его полную перезагрузку.

При необходимости, например выполнение технического обслуживания ПО, приостановка его работы и т.д., отслеживание пульсирующего сигнала можно приостановить через Web-интерфейс настройки программного модуля.

На основном вычислительном модуле при запуске скрипта с пайплайном (*run_pipeline_db.py*) выполняется периодичный процесс подачи напряжения на выбранный GPIO.