

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

**ПРОГРАММНЫЕ МОДУЛИ ОБЕСПЕЧЕНИЯ ПРОГРАММНО-
АППАРАТНЫХ КОМПЛЕКСОВ В ООО «ДИВИЛАЙН»**

ОТЧЕТ

ПО РЕЗУЛЬТАТАМ

производственной практики: «Научно-исследовательская работа»

(вид практики: тип практики)

Обучающийся гр. 430-2

(подпись) А.А.Лузинсан
(И.О. Фамилия)

(дата)

Руководитель практики от профильной
организации:

ведущего инженера-программиста
ООО «ДиВиЛайн»
(должность, ученая степень, звание)

оценка М.П. _____
(подпись) К.А.Рылов
(И.О. Фамилия)

(дата)

Руководитель практики от
Университета: профессор каф. АСУ, к.
юр. наук, PhD
(должность, ученая степень, звание)

оценка _____
(подпись) С.М.Левин
(И.О. Фамилия)

(дата)

Томск 2023

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизированных систем управления (АСУ)

УТВЕРЖДАЮ
Зав. кафедрой АСУ
Романенко В.В.

«__» _____ 2023 г.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

на производственную практику: «Научно-исследовательская работа»

студенту гр. 430-2 факультета систем управления

Лузинсан Анастасии Александровне

(Ф.И.О студента)

- 1. Тема практики:** программные модули обеспечения программно-аппаратных комплексов в ООО «Дивилайн».
- 2. Цель практики** – систематизация, закрепление и расширение теоретических и практических знаний по технологии проектирования и разработке программных средств с использованием современных компьютерных технологий на основе системного анализа проблемной ситуации.
- 3. Сроки прохождения практики:** с 6 февраля по 4 марта 2023 г.

Совместный рабочий график (план) проведения практики

№ п/п	Перечень заданий	Сроки выполнения
1.	Прохождение техминимума и тех. безопасности на рабочем месте. – Ознакомление со структурой предприятия, видами деятельности, процессами организации управления деятельностью предприятия; – Изучение целей и функций автоматизации технологических процессов, автоматизированных систем управления, используемых средств вычислительной техники в действиях данного предприятия.	06.02.2023
2.	– Расширение навыков по применению методов автоматизации производственных процессов. Изучение и освоение классов библиотеки Qt5.	08.02.2023

3.	– Индивидуальное задание от предприятия: – «Разработка модуля для отображение прогресса выполнения измерений».	10.02.2023
4.	– Индивидуальное задание от предприятия: – «Разработка модуля управления транслятором».	17.02.2023
5.	– Изучение руководства пользователя и руководства разработчика фильтра цифровой видеолaborатории «DiViLab»; – Обзор цифровой видеолaborатории DiViLab;	20.02.2023
3.	– Изучение технологии обработки информации с видеопотока, алгоритмов решения задач по компьютерному зрению с применением библиотеки OpenCV.	21.02.2023
5.	– Индивидуальное задание от предприятия: «Разработка алгоритма измерения размера поля зрения камеры в условных единицах по таблице "Шахматная доска"».	27.02.2023
6.	Написание отчета по практике, заполнение дневника практики и создание презентации с основными результатами практики.	03.03.2023

Дата выдачи: «06» февраля 2023 г.

Руководитель практики от университета
профессор каф. АСУ, к. юр. наук, PhD
(должность)

(Подпись)

С.М.Левин
(ФИО)

Согласовано:

Руководитель практики от профильной
организации

ведущий инженер-программист
ООО «ДиВиЛайн»
(должность)

(Подпись)
МП

К.А.Рылов
(ФИО)

Принял к исполнению «06» февраля 2023 г.

Студент гр. 430-2

(Подпись)

А.А.Лузинсан
(ФИО)

Оглавление

Введение	5
1 ОСНОВНАЯ ЧАСТЬ.....	7
1.1 Анализ деятельности предприятия	7
1.2 Выбор стека технологий.....	8
1.3 Обзор предметной области.....	10
1.4 Постановка задачи.....	11
1.4.1 Модуль отображения прогресса выполнения измерений.....	11
1.4.2 Модуль управления трансляторами.....	12
1.4.3 Алгоритм измерения размера поля зрения камеры	14
1.5 Обзор аналогов	15
1.5.1 Модуль отображения прогресса выполнения измерений.....	15
1.5.2 Модуль управления трансляторами.....	16
1.5.3 Алгоритм измерения размера поля зрения камеры	16
1.6 Индивидуальные задания	16
1.6.1 Модуль отображения прогресса выполнения измерений.....	16
1.6.2 Модуль управления трансляторами.....	18
1.6.3 Алгоритм измерения размера поля зрения камеры	20
Заключение.....	22
Список использованных источников.....	23

Введение

В данной работе рассматривается проектирование программных модулей с использованием библиотеки Qt5, а также разработка алгоритма с использованием библиотеки OpenCV. Данные модули предназначены для автоматизации взаимодействия с программно-аппаратным комплексом. В свою очередь алгоритм предоставляет пользователю необходимую информацию в рамках задачи.

Цель работы состоит в реализации виджетов, которые будут внедрены в программно-аппаратный комплекс, а также реализации алгоритма измерения размера поля зрения камеры.

Программный модуль — это описание процесса, оформляемое как самостоятельный программный продукт, пригодный для использования в описаниях процесса. Это означает, что каждый программный модуль программируется, компилируется и отлаживается отдельно от других модулей программы, и тем самым, физически разделен с другими модулями программы. Более того, каждый разработанный программный модуль может включаться в состав разных программ, если выполнены условия его использования, декларированные в документации по этому модулю. Таким образом, программный модуль может рассматриваться и как средство борьбы со сложностью программ, и как средство борьбы с дублированием в программировании [1].

Компетенции:

- ОПК-1 – способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности;

- ОПК-3 – способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;

- ОПК-5 – способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем;
- ОПК-6 – способен разрабатывать бизнес-планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием;
- ОПК-7 – способен участвовать в настройке и наладке программно-аппаратных комплексов;
- ОПК-9 – способен осваивать методики использования программных средств для решения практических задач;
- УК-1 – способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач;
- УК-4 – способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(ых) языке(ах);
- УК-5 – способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах;
- УК-7 – способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности;
- УК-8 – способен создавать и поддерживать в повседневной жизни и в профессиональной деятельности безопасные условия жизнедеятельности для сохранения природной среды, обеспечения устойчивого развития общества, в том числе при угрозе и возникновении чрезвычайных ситуаций и военных конфликтов.

1 ОСНОВНАЯ ЧАСТЬ

1.1 Анализ деятельности предприятия

Научно-производственное предприятие ДиВиЛайн является аккредитованной ИТ-организацией, осуществляющая комплексные инновационные разработки, связанные с обработкой и анализом изображений, видеоданных и других сигналов [2].

ДиВиЛайн разрабатывает системы технического зрения для промышленных применений. Такие системы могут иметь различную архитектуру и удаленность от центра управления.

Компетенции ДиВиЛайн [3]:

- анализ, обработка, передача изображений и видеоданных, промышленное телевидение, измерения, автоматизация;
- создание и настройка серверов, применение систем виртуализации;
- применение технологий ускорения вычислений (CUDA, OpenGL...);
- разработка мобильных вычислительных платформ (железо: DSP, микроконтроллеры);
- специализированные видеодатчики, позволяющие расширить применимость видеосистем;
- машинное обучение, анализ данных (предиктивный анализ, автоматизация, управления тех. процессами);
- базы данных, интерфейсное ПО взаимодействия с базами данных;
- клиент-серверные решения, разработка программного обеспечения для систем на базе PC (C++, Qt, Python, ...), кроссплатформенные решения;
- мобильные приложения;
- разработка технической документации (ЕСКД/ЕСПД).

На счете НПП ДиВиЛайн ряд успешно выполненных научно-исследовательских и опытно-конструкторских работ, таких как:

- программно-технические комплексы для проведения

криминалистических исследований видеозаписей.

- базы данных образцов записей, диагностических признаков видеоредакторов и видеоаппаратуры, специальное программное обеспечение для работы с базами данных.
- программные комплексы исследования видеозаписей, выявления фактов монтажа, определения средств создания видеофайлов.

1.2 Выбор стека технологий

В качестве языка программирования на производстве используется язык C++, так как он оказывается значительно быстрее во многих задачах компьютерного зрения [4].

Qt5 выступает в качестве основной библиотеки для разработки как виджетов, так и фильтра, которые будут внедрены в программно-аппаратный комплекс. Qt содержит полный набор интуитивно понятных и модульных классов C++ и загружается с API-интерфейсами для упрощения разработки приложений. Qt создает легко читаемый, поддерживаемый и повторно используемый код с высокой производительностью во время выполнения и небольшими размерами – и он кроссплатформенный [5].

OpenCV, которая была выбрана для реализации поставленных задач, является кроссплатформенной, широко используемой библиотекой компьютерного зрения с открытым исходным кодом. Основными преимуществами OpenCV являются вычислительная эффективность с акцентом на приложения, работающие в реальном времени [6].

DiViLab является инструментом исследования и разработки алгоритмов для систем технического зрения, подготовки специалистов в сфере видеоинформационных технологий, которая обеспечивает следующее [7]:

- изучение форматов файлов и современных стандартов сжатия видеоданных;
- проведение практических занятий по основам обработки изображений и

видеоданных;

- обучение разработчиков алгоритмов обработки и анализа изображений, включая программирование на языке C++;
- выполнение научных исследований и постановка экспериментов;
- ускорение создания и оптимизация новых алгоритмов, предназначенных для встраиваемых систем и различного оборудования;
- обучения разработчиков с использованием известных библиотек и современных технологий, таких как OpenCV, FFTW, CUDA, OpenGL и т.д.

TortoiseSVN [8] - это клиент Apache Subversion (SVN), реализованный как расширение оболочки Windows. Он интуитивно понятен и прост в использовании, поскольку для его запуска не требуется клиент командной строки Subversion. Он разработан под лицензией GPL, что означает, что он абсолютно бесплатен для любого пользователя, в том числе в коммерческой среде, без каких-либо ограничений.

Redmine [9] — это гибкое веб-приложение для управления проектами. Он написан с использованием фреймворка Ruby on Rails и является кроссплатформенным. Redmine имеет открытый исходный код и выпущен под лицензией GNU. Некоторые из основных особенностей Redmine являются:

- поддержка нескольких проектов;
- гибкое управление доступом на основе ролей;
- гибкая система отслеживания задач;
- диаграмма Ганта и календарь;
- управление новостями, документами и файлами;
- уведомления по электронной почте;
- вики проекта;
- наличие форумов проекта;
- отслеживание времени выполнения задач;
- интеграция с SCM (SVN, CVS, Git, Mercurial и Bazaar);
- мультиязычная поддержка и многое другое.

1.3 Обзор предметной области

Линейный транслятор - устройство для точного перемещения объекта вдоль прямой линии. В его конструкцию входит подвижная платформа и неподвижное основание, которые соединены системой подшипников. На верхней поверхности платформы предусмотрены специальные монтажные отверстия (М6 и/или М4) для закрепления оптических элементов или оборудования. Положение контролируется с помощью прецизионного привода, например, регулировочного винта или микрометра.

Горизонтальное перемещение вдоль одной оси реализуется при использовании позиционеров X-типа. Большинство таких устройств обладают свойством модульности. Это означает, что несколько линейных трансляторов X-типа могут быть объединены (напрямую или при помощи кронштейнов) в многоосевую систему:

- XY;
- XZ;
- XYZ.

Для чисто вертикального перемещения необходимы модели Z-конфигурации или вертикальные позиционеры.

Поворотные платформы используются для плавного вращения оптических элементов, например, поляризаторов и волновых пластинок. Они позволяют реализовать грубое вращение в пределах 360° с точностью 1° , а также точное вращение в небольших пределах (порядка 10°) при помощи микрометров.

Стандартные позиционеры можно оптимизировать для работы в вакууме до 10^{-6} Торр или условиях низких температур [10].

Линейный транслятор на шарикоподшипнике применяется для научных исследований, автоматического измерения и испытаний.

1.4 Постановка задачи

На предприятии было выдано несколько задач, среди которых была разработка виджетов и создание алгоритма.

Первые две задачи предполагались для расширения навыков по применению методов автоматизации производственных процессов, а также разработке виджетов на языке C++, изучению и освоению классов библиотеки Qt5. Последняя же задача была направлена на исследование и анализ существующих алгоритмов для выборки наилучшей альтернативы.

1.4.1 Модуль отображения прогресса выполнения измерений

Данный модуль должен выступать визуальным отображением прогресса выполнения измерений. Примерный интерфейс модуля представлен на рисунке 1.2.

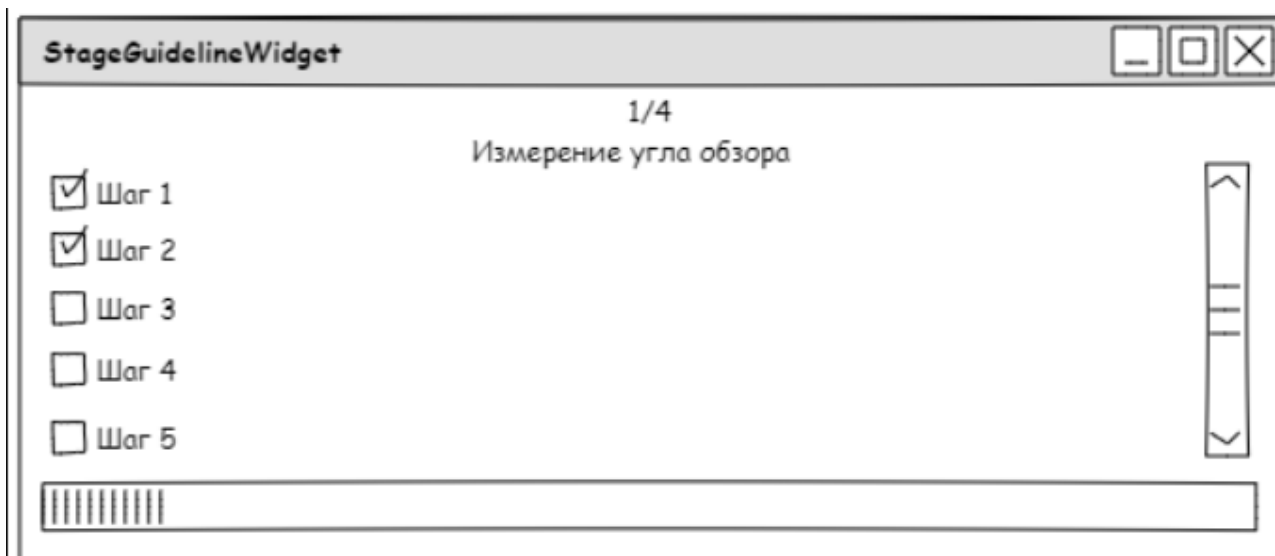


Рисунок 1.2 – Интерфейс модуля отображения прогресса выполнения измерений

Для реализации данного модуля требовалось определить следующие методы:

- `addGuideline(name)` – слот, который добавляет название руководства в конец списка руководств;
- `setGuidelines(names)` – слот, заменяющий список руководств новым списком;

- `addStages(guideline_name, stage_name)` – слот, добавляющий к указанному руководству ещё один шаг;
- `setStages(guideline_name, stage_names)` – слот, устанавливающий новый список шагов указанному руководству;
- `selectGuideline(name)` – слот, который позволяет отобразить указанное руководство;
- `markStage(name, mark)` – слот для пометки указанного шага `name` значением `mark` в текущем руководстве;
- `clear()` – слот, позволяющий очистить список всех шагов и всех руководств;
- `guidelineCount()` – метод, возвращающий количество хранящихся руководств;
- `stagesCount()` – метод, возвращающий количество всех шагов текущего руководства.

1.4.2 Модуль управления трансляторами

Данный модуль должен предоставлять возможность взаимодействия с транслятором. Взаимодействие между программным модулем и драйвером осуществляется через механизм сигналов и слотов. Таким образом от модуля драйверу посылается сигнал смещения на новую позицию относительно старой. Интерфейс виджета изображён на рисунке 1.3.

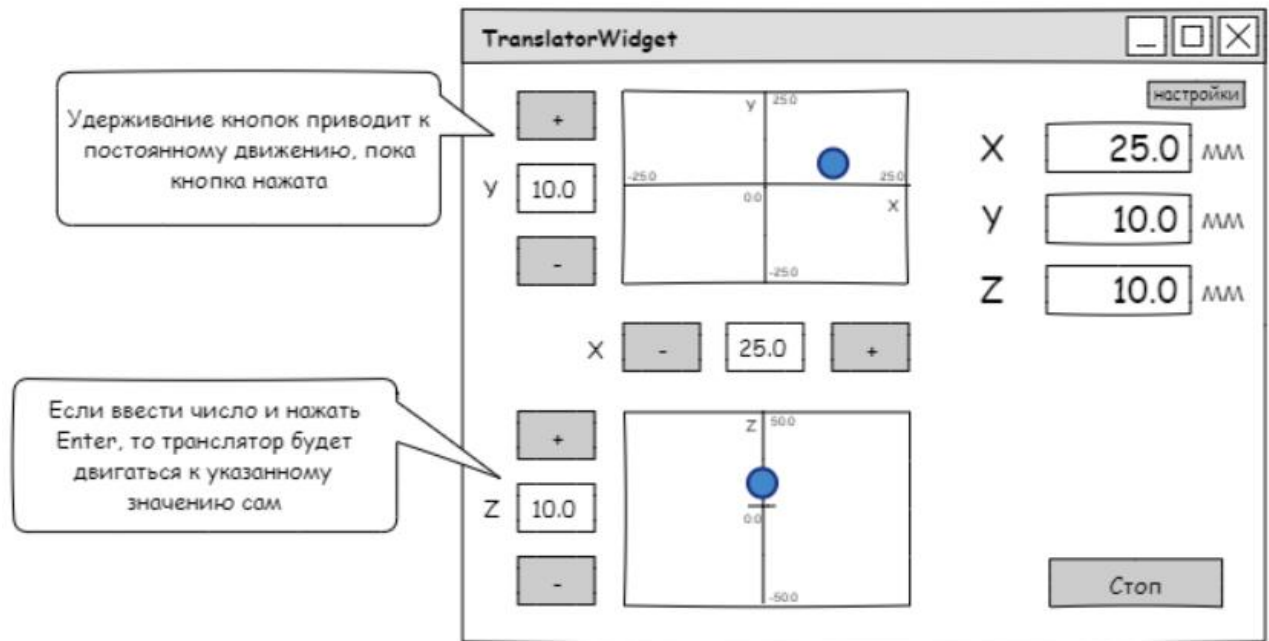


Рисунок 1.3 – Интерфейс модуля управления трансляторами

Взаимодействие с драйвером транслятора осуществляется посредством следующий методов:

- `getPosition()` – позволяет получить текущую позицию транслятора;
- `setPosition(Position)` – устанавливает текущее положение транслятора.

Транслятор в свою очередь возвращает код операции;

- `move(delta_x, delta_y, delta_z)` – сдвигает текущее положение транслятора на новое относительно старого. Транслятор возвращает код операции;

- `waitUntilMoves()` – устанавливает транслятор в режим ожидания до следующего возникшего сигнала;

- `stop()` – приостанавливает движение транслятора;

- `getMinPosition()` – возвращает значение минимальной позиции транслятора;

- `getMaxPosition()` – возвращает значение максимальной позиции транслятора;

- `lockWidget()` – приостанавливает возможность взаимодействия с

транслятором;

- `unlockWidget()` - возобновляет возможность взаимодействия с транслятором.

В ходе выполнения задачи требовалось реализовать следующие методы:

- `setRangePosition(min_Position, max_Position)` – слот, который устанавливает новую допустимую область значений позиции транслятора;

- `setPosition(Position)` – слот, который обновляет график, отображающий текущее положение транслятора, а также панель отображения значений текущих координат;

- `lock()` – слот, блокирующий пользовательское управление устройством, но не блокирует индикацию, отображение текущего положения, что не влияет на работу устройства;

- `unlock()` – слот, восстанавливающий пользовательское управление устройством;

- `stopped()` – сигнал, уведомляющий об остановке взаимодействия с транслятором (при нажатии кнопки «Стоп»);

- `moved(delta_x, delta_y, delta_z)` – сигнал, который посылается при пользовательском изменении позиции транслятора посредством кнопок виджета;

- `settingsChanged(settings)` – сигнал, уведомляющий об изменении настроек транслятора.

1.4.3 Алгоритм измерения размера поля зрения камеры

В последней задаче необходимо было исследовать и проанализировать существующие алгоритмы на предмет наименее трудозатратного. Тестовое изображение представляет собой таблицу «Шахматная доска», которая представлена на рисунке 1.4. Описание задачи представлено ниже:

«Необходимо разработать алгоритм, который позволит получить ширину и высоту наблюдаемого в поле зрения камеры изображения шахматной доски в условных единицах. В данном случае условной единицей является размер среднего квадрата (черного или белого). На вход алгоритм получает изображение с камеры, а на выход отдает ширину и высоту в "квадратах" (высота: 4.65, ширина: 8.32).»

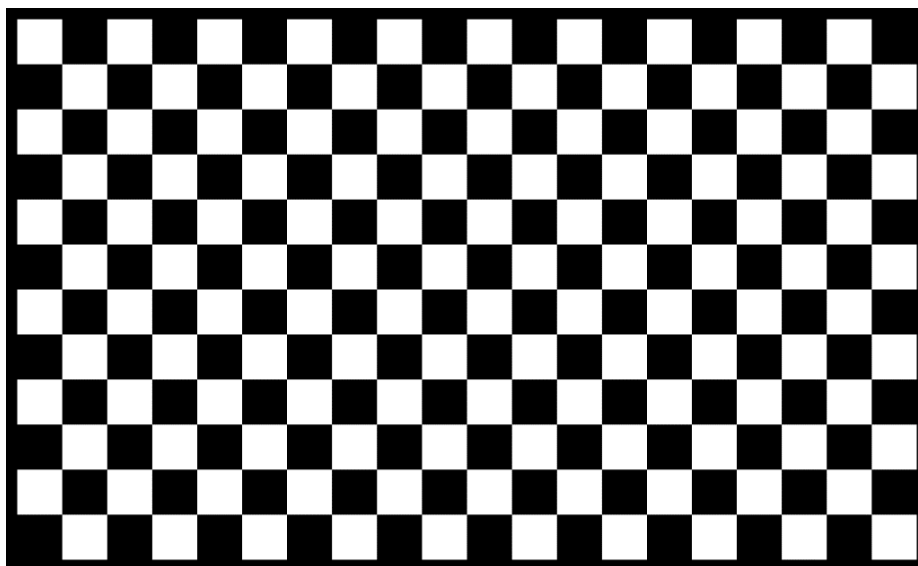


Рисунок 1.4 – Тестовое изображение «Шахматная доска»

Первым делом при анализе задачи всплывало решение, которое касалось обнаружения всех квадратов на изображении. После подсчёта сторон каждого такого квадрата можно было бы сказать сторону среднего квадрата.

1.5 Обзор аналогов

1.5.1 Модуль отображения прогресса выполнения измерений

На этапе реализации области прокрутки для чекбоксов была обнаружена проблема [11], заключающаяся в отсутствии автопереноса текста, принадлежащих чекбоксам. В ходе исследования были найдены реализации пользовательских классов чекбокса, где данное недоразумение устранено. Таким образом в процессе реализации модуля отображения прогресса выполнения измерений было интегрировано решение, устраняющее данную проблему [12].

1.5.2 Модуль управления трансляторами

Реализация данного модуля предполагала интеграцию графика, содержащего центрированные оси. Однако, существующая в данный момент библиотека QtCharts позволяет выравнивать оси только в краю сцены, тогда как центрирование там явно запрещено [13]. В связи с чем были исследованы альтернативы таких графиков, среди которых была найдена библиотека Qwt [14] и QCustomPlot [15]. После нескольких безуспешных попыток установки вместе с комплектом, предоставляемым предприятием, было принято решение собственноручно реализовать только тот функционал, который требовался для модуля [16].

1.5.3 Алгоритм измерения размера поля зрения камеры

На официальном ресурсе OpenCV предоставлена демонстрационная программа для нахождения квадратов в списке изображений, которая использует пирамидальное масштабирование, детектор граней Canny и упрощение контуров. Она возвращает последовательность квадратов, обнаруженных на изображении, которую в дальнейшем можно проанализировать и выявить среднюю сторону квадрата [17]. Однако такой подход был гораздо трудозатратнее и дольше, что не подходило под требования задачи.

Также, в репозитории пользователя alyssaq на github уже собраны различные реализации этой задачи [18]. Однако все они сводились к использованию тяжелых решений через поиск граней и задеирования Canny.

1.6 Индивидуальные задания

1.6.1 Модуль отображения прогресса выполнения измерений

Первой полученной задачей было создание виджета, отображающего прогресс выполнения измерения. В ходе реализации данного виджета были выполнены следующие подпункты:

- каркас окна;
- добавление progress bar, добавление guideline'a, списка шагов;
- реализация методов addGuideline, addStage, markStage, setGuidelines, setStages;
- расширение до списка гайдлайнов с добавлением шагов в указанному гайдлайну;
- переключение между виджетами checkboxes
- автоматическое переключение на новый guideline при выполнении всего списка;
- реализация метода clear();
- реализация guidelinesCount(): int и stagesCount(): int

Далее, после ревью кода последовало изменение логики виджета. Итак, были внесены следующие правки:

- реализация updateContent;
- динамическое добавление checkboxes в scrollarea в updateContent;
- методы addStage, setStages, addGuideline, setGuidelines selectGuideline, markStages переделаны;
- метод indexOfGuideline для текущего индекса гайдлайна из Map'a;
- метод stagesAllCount;
- добавление Qt::WA_DeleteOnClose
- исправление making QCheckBox readonly, but not grayed-out;
- настройка scrollWidget (setSpacing, setMargin, setAlignment);
- отладка, фикс багов

После принятия изменений была выдана подзадача по добавлению автоматической прокрутки scroll'a и выделение следующего шага:

- создание класса QCustomCheckBox со встроенным QLabel для wrapWord;
- реализация автопереноса текста;

- автоматическая прокрутка scroll'a и выделение следующего шага
- тестирование реализации

Таким образом полученный виджет представлен на рисунке 1.5.

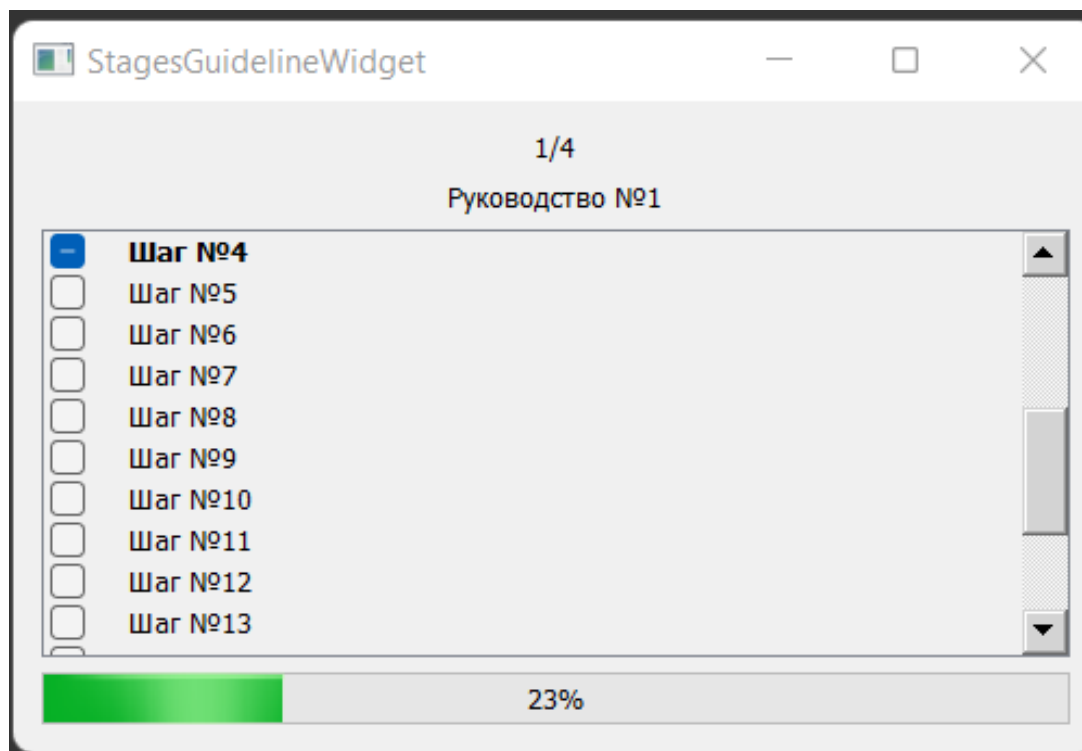


Рисунок 1.5 – Модуль отображения выполнения измерений

1.6.2 Модуль управления трансляторами

В ходе реализации данного виджета задача была разбита на две дочерние подзадачи: реализация общего класса `TranslatorWidget` и реализацию класса `TranslatorGraphics`, отвечающего за график отображения транслятора.

Подзадачи задачи `TranslatorWidget` состояли в следующем:

- реализация слотов `setRangePosition`, `setPosition`, `lock` и `unlock`
- добавление connect для кнопок (удерживание) и `lineEdit` ("Enter")
- правка логики: обновление графика и полей отображения координат только при `setPosition`
 - при взаимодействии с `button` и `lineEdit` меняются только `lineEdit` и посылается сигнал на изменение транслятору
 - переделано под `gridLayout`

- добавление окна настроек
- реализация методов создания и применения настроек
- фиксация положения кнопок и lineEdit

Подзадачи задачи TranslatorGraphics содержали следующее:

- настройка каркаса графика, добавление осей, label осей, граничных значений
- добавление таймера для обновления положения элементов графика
- размещение эллипса на сцене в соответствии с шагом осей
- реализация слотов setZeroPoint, setPoint, setValuePoint, setRangeAxis для двумерного графика транслятора и private методов updateAxes, updateCircle
- настройка эллипса (закрашивание, добавление прозрачности, изменение радиуса в зависимости от высоты графика).

Итоговый модуль управления трансляторами представлен на рисунке 1.6.

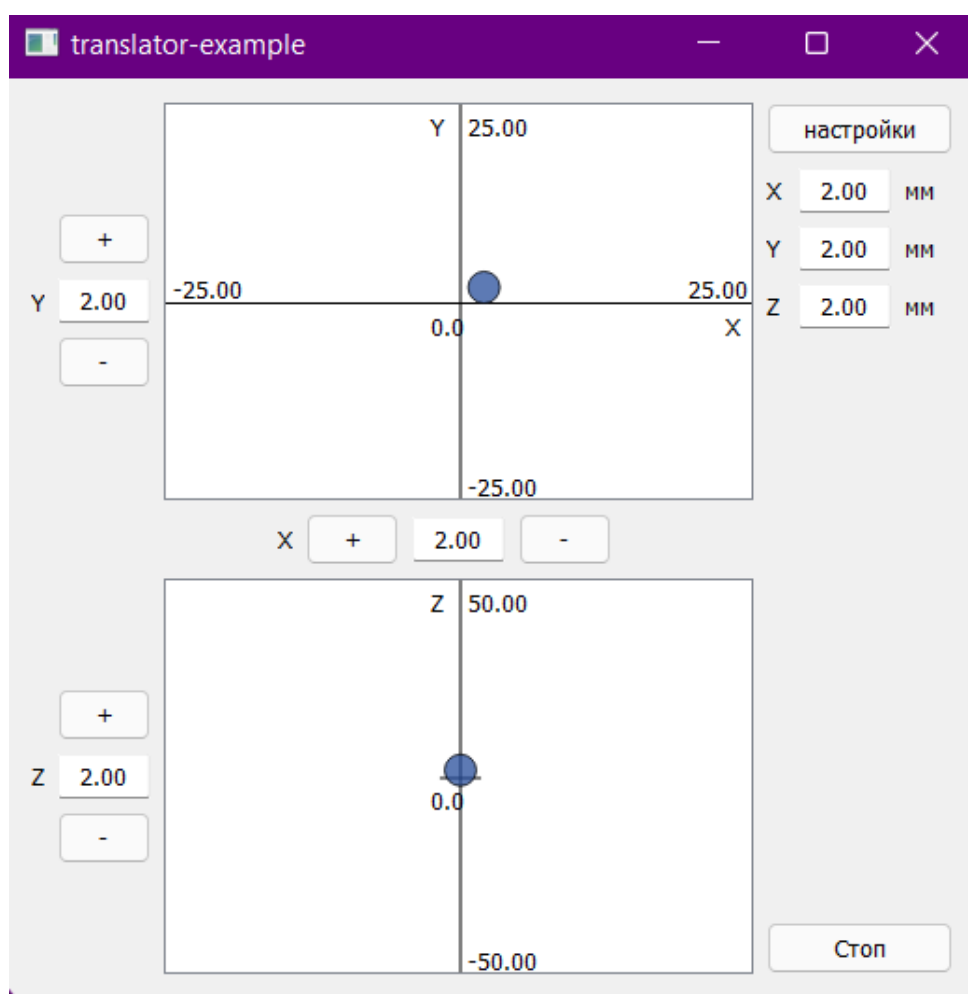


Рисунок 1.6 - Модуль управления трансляторами

1.6.3 Алгоритм измерения размера поля зрения камеры

После внедрения и анализа различных решений, были конкретизированы требования поставленной задачи:

1. Исходная таблица, поступающая в виде изображения шахматной доски расположена прямо и без искажений;
2. Предотвращены случаи поворота камеры;

Исходя из вышеизложенных требований задача на порядок упростилась, следовательно обнаруженные методы решения были бы излишни. Предложенный вариант представляет собой конвертацию изображения из формата RGB в формат BGR, используемый классом Mat библиотеки OpenCV посредством вызова функции `ConvertRGB24ToOpencvBGR24`, определённой в фильтре. Далее запускается простая предобработка изображения в чёрно-белый формат методом `cv::cvtColor(image, image, cv::COLOR_BGR2GRAY)`. И наконец, изображение бинаризуется с использованием метода `cv::threshold()` с параметром `cv::THRESH_BINARY`. После предобработки изображения запускается алгоритм поиска средней длины квадрата в таблице.

Перебирая отдельную строку (столбец) изображения, создаётся вектор длин линий одинакового цвета. Проверяя каждый раз итератор, указывающий на очередной пиксель в строке (столбце), он сравнивается с предыдущим обнаруженным пикселем, и если они не совпадают, то линия считается завершённой и следовательно счётчик линии заносится в вектор линий, счётчик длины линии обнуляется, а предыдущим цветом становится текущий. По выходу из цикла проверяется, были найдены ли линии. Если да, то, согласно условиям съёмки, удаляются первый и последний элементы и возвращается средняя длина линии, в противном случае возвращается -1.

После прохода по строкам в переменной класса фильтра запоминается средняя высота условной единицы – «шахматная ячейка» как среднее арифметическое из всех подсчитанных строк. Аналогичная ситуация происходит и со столбцами, с одним условием, что запоминается средняя ширина условной

единицы как среднее арифметическое из всех подсчитанных столбцов.

Итоговым решением задачи считается вычисление выделенной области как отношение ширины и высоты выделенной области на соответствующее значение ячейки в пикселях. Демонстрация решения предоставлена на рисунке 1.7.

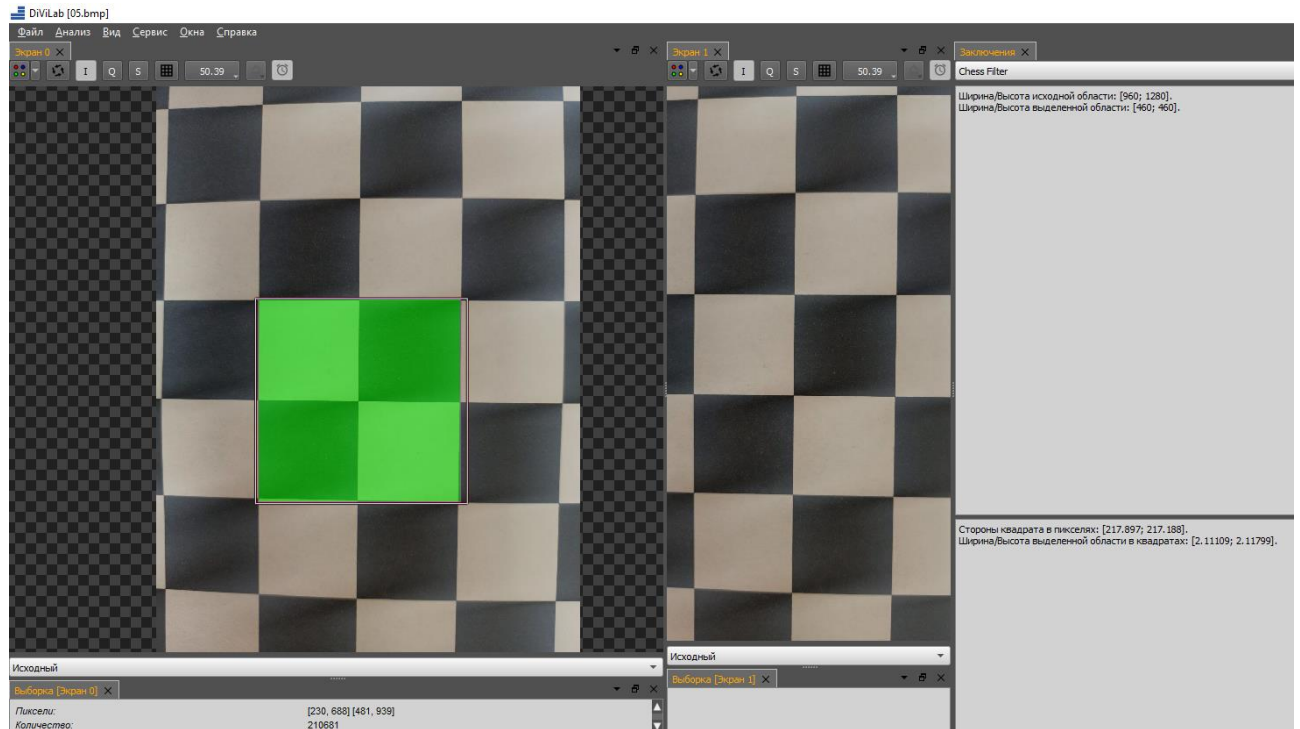


Рисунок 1.7 – Результат работы алгоритма измерения размера поля зрения камеры

Заключение

При прохождении производственной практики «Научно-исследовательская работа» были получены результаты, приведенные ниже:

- изучена предметная область;
- определена цель работы и сделан разбор задач;
- осуществлен сравнительный обзор существующих аналогов, выявлены их недостатки;
- изучена цифровая видеолaborатория DiViLab, основы библиотеки OpenCV, а также закреплены навыки, касаемые языка программирования C++ и библиотеки Qt в среде QtCreator.
- разработаны модули и алгоритм измерения размера поля зрения камеры;
- осуществлена программная реализация и тестирование разработанных программных продуктов;
- показана его работоспособность и эффективность.

Список использованных источников

1. РАЗРАБОТКА СТРУКТУРЫ ПРОГРАММЫ И МОДУЛЬНОЕ ПРОГРАММИРОВАНИЕ [Электронный ресурс]: статья на портале Bourabai. URL: <http://bourabai.ru/alg/pro07.htm> (дата обращения: 06.02.2023).
2. Дивилайн – российское научно-производственное предприятие [Электронный ресурс]: официальная страница предприятия Дивилайн. URL: <https://diviline.ru/about-diviline/> (дата обращения: 06.02.2023).
3. Видеоаналитика, промышленная автоматизация [Электронный ресурс]: официальная страница предприятия Дивилайн. URL: <https://diviline.ru/about-diviline/about-automation/> (дата обращения: 06.02.2023).
4. Сравниваем скорость C++ и Python на примере стереозрения в OpenCV на Raspberry Pi [Электронный ресурс]: статья на Proglib. URL: <https://proglib.io/p/opencv-sravniваем-skorost-koda-c-i-python-na-raspberry-pi-stereozrenie-2020-02-06> (дата обращения: 07.02.2023).
5. Qt Framework [Электронный ресурс]: документация Qt. URL: <https://www.qt.io/product/framework> (дата обращения: 08.02.2023).
6. OpenCV with Qt [Электронный ресурс]: документация Qt. URL: https://wiki.qt.io/OpenCV_with_Qt (дата обращения: 08.02.2023).
7. DiViLab — цифровая видеолaborатория [Электронный ресурс]: Описание продукта предприятия Дивилайн. URL: <https://diviline.ru/products/divilab/> (дата обращения: 09.02.2023).
8. About TortoiseSVN [Электронный ресурс]: официальный сайт TortoiseSVN. URL: <https://tortoisesvn.net/> (дата обращения: 20.02.2023).
9. Redmine [Электронный ресурс]: официальный сайт Redmine. URL: <https://www.redmine.org/> (дата обращения: 20.02.2023).
10. Линейные трансляторы [Электронный ресурс]: каталог магазина lenlasers. URL: <https://lenlasers.ru/catalog/lineynye-translyatory/> (дата обращения: 20.02.2023).
11. Add support of word wrap text in QCheckBox [Электронный ресурс]:

отчёт о баге Qt. URL: <https://bugreports.qt.io/browse/QTBUG-5370> (дата обращения: 10.02.2023).

12. qtCustomPlugins [Электронный ресурс]: репозиторий плагина qtCustomPlugins. URL: <https://github.com/ThiBsc/qtCustomPlugins> (дата обращения: 13.02.2023).

13. QtCharts Axis Alignment Center [Электронный ресурс]: вопрос на StackOverflow. URL: <https://stackoverflow.com/questions/55116545/qtcharts-axis-alignment-center> (дата обращения: 13.02.2023).

14. Qwt - Qt Widgets for Technical Applications [Электронный ресурс]: Официальный ресурс библиотеки Qwt. URL: <https://qwt.sourceforge.io/> (дата обращения: 14.02.2023)

15. Qt Plotting Widget - Introduction [Электронный ресурс]: Официальный ресурс библиотеки QCustomPlot. URL: <https://www.qcustomplot.com/> (дата обращения: 15.02.2023)

16. Qt/C++ - Урок 017. QGraphicsScene или как работать с графикой в Qt [Электронный ресурс]: Урок на портале Evileg. URL: <https://evileg.com/ru/post/80/> (дата обращения: 16.02.2023)

17. samples/cpp/squares.cpp [Электронный ресурс]: документация OpenCV. URL: https://docs.opencv.org/5.x/db/d00/samples_2cpp_2squares_8cpp-example.html (дата обращения: 21.02.2023)

18. alyssaq/opencv [Электронный ресурс]: Репозиторий пользователя alyssaq. URL: <https://github.com/alyssaq/opencv/blob/master/squares.cpp> (дата обращения: 23.02.2023)