

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ РАДИОФИЗИКИ И КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ**

Кафедра телекоммуникаций и информационных технологий

БРИКСА

Артём Андреевич

**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ КОНТРОЛЯ
АВТОМАТИЗИРОВАННОЙ СБОРКИ МОДУЛЕЙ SMS ДЕТЕКТОРА
БОЛЬШОГО АДРОННОГО КОЛЛАЙДЕРА**

Дипломная работа

Научный руководитель:
кандидат технических наук,
доцент И.П. Стецко

Допущена к защите

«__» _____ 2017 г.

Зав. кафедрой телекоммуникация и информационных технологий,
кандидат физико-математических наук Ю.И. Воротницкий

Минск, 2017

Оглавление

РЕФЕРАТ	3
Введение.....	7
Глава 1 Сборка модулей	11
1.1 Ручная сборка с помощью механической платформы	11
1.2 Система автоматизированной сборки	12
1.3 Сборочная платформа.....	14
1.4 Быстросохнущее клеящее вещество	16
1.5 Процесс автоматизированной сборки	19
Глава 2 Программное обеспечение контроля сборки	21
2.1 Общая структура приложения	21
2.1.1 Шаблон проектирования «Модель-Представление-Контроллер» 21	
2.1.2 Библиотека OpenCV	24
2.2 Распознавание образов	25
2.2.1 Предварительная обработка изображения, полученного с камеры 26	
2.2.2 Определение позиции и планарной ориентации маркера	27
2.2.3 Определение приблизительного положения следующего маркера и/или конечное вычисление планарной ориентации сенсора	29
2.3 Функционал приложения	30
Глава 3 Тестирование системы и сборка первого прототипа.....	33
3.1 Тест точности процедуры распознавания образов	33
3.1.1 Распознавание образов на угле стеклянного образца	34
3.1.2 Распознавание образов на маркере фиктивного сенсора	37
3.2 Тест захвата образца	38
3.2.1 Тест захвата образца без помощи сборочной платформы	38
3.2.2 Тест захвата образца с помощью сборочной платформой	40
3.3 Сборка первого прототипа	41
Заключение	45
Список использованных источников	46
Приложение	47

РЕФЕРАТ

Дипломная работа, 52 страницы, 33 рисунков (схема, диаграммы, графики, гистограммы, фотографии, рисунки), 2 приложения, 12 источников.

АВТОМАТИЗАЦИЯ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ, ИНСТРУМЕНТ РАЗРАБОТКИ QT, БИБЛИОТЕКА OPENCV, ШАБЛОН ПРОЕКТИРОВАНИЯ MVC, ОБРАБОТКА ИЗОБРАЖЕНИЙ, ТЕСТИРОВАНИЕ, ВТОРАЯ СТАДИЯ МОДЕРНИЗАЦИИ CMS ТРЕКЕРА.

Объект исследования – система автоматизированной сборки модулей CMS детектора Большого адронного коллайдера (БАК), в частности – программное обеспечение этой системы.

Цели работы – модернизировать программное обеспечение контроля автоматизированной сборки модулей CMS детектора БАК второй стадии модернизации, модернизация непрограммных элементов системы автоматизированной сборки, проведение тестов системы, автоматизированная сборка первого прототипа модуля.

Трекер CMS детектора БАК второй стадии модернизации состоит из двух типов модулей (PS и 2S), каждый состоящий из двух кремневых сенсоров, необходимой для их работы электроники и других элементов корпуса. В случае PS модуля, сенсоры должны быть собраны с относительной точностью около 40 мкм. Чтобы выполнить данные требования и сократить время на сборку, предлагается использовать систему автоматизированной сборки. Данная система основывается на высокоточной двигательной установке с интегрированной в ней камерой высокого разрешения, которая позволяет осуществлять распознавание образов, систему фиксации отрицательным давлением и специального программного обеспечения для контроля их работы.

С целью контроля всей системы автоматизированной сборки было разработано специальное программное обеспечение для ПК, представляющее из себя Qt приложение. Всё необходимое аппаратное обеспечение подключается к ПК по интерфейсу USB. Контроль над всеми элементами системы интегрирован в одно приложение, именуемым PSAuto (от автоматизированной сборки PS модуля). Приложение написано на языке программирования C++ и использует Qt фреймворк версии 4.8.7. Функционал приложения позволяет осуществлять полный спектр процедур автоматизированной сборки, её контроля и тестирование всей системы.

РЭФЕРАТ

Дыпломная праца, 52 старонак, 33 малюнкаў (схемы, дыяграмы, графікі, гістаграмы, фатаграфіі, малюнкi), 2 дадатка, 12 крыніц.

АЎТАМАТЫЗАЦЫЯ, ПРАГРАМНАЕ ЗАБЕСПЯЧЭННЕ, ИНСТРУМЕНТ РАСПРАЦОЎКІ QT, БІБЛІЯТЭКА OPENCV, ШАБЛОН ПРАЕКТАВАННЯ MVC, АПРАЦОЎКІ МАЛЮНКАЎ, ТЭСЦІРАВАННЕ, ДРУГАЯ СТАДЫЯ МАДЭРНІЗАЦЫІ CMS ТРЭКЕРА.

Аб'ект даследавання – сістэма аўтаматызаванай зборкі модуляў CMS дэтэктара Вялікага адроннага калайдэра (ВАК), у прыватнасці – праграмнае забеспячэнне гэтай сістэмы.

Мэты працы – мадэрнізаваць праграмнае забеспячэнне кантролю аўтаматызаванай зборкі модуляў CMS дэтэктара ВАК другой стадыі мадэрнізацыі, мадэрнізацыя непрограммных элементаў сістэмы аўтаматызаванай зборкі, правядзенне тэстаў сістэмы, аўтаматызаваная зборка першага прататыпа модуля.

Трэкер CMS дэтэктара ВАК другой стадыі мадэрнізацыі складаецца з двух тыпаў модуляў (PS і 2S), кожны з якіх складаецца з двух крэмневых сэнсараў, неабходнай для іх працы электронікі і іншых элементаў корпуса. У выпадку PS модуля, сэнсары павінны быць сабраны з адноснай дакладнасцю каля 40 мкм. Каб выканаць дадзеныя патрабаванні і скараціць час на зборку, прапануецца выкарыстоўваць сістэму аўтаматызаванай зборкі. Дадзеная сістэма засноўваецца на высокадакладнай рухальнай ўсталёўцы з інтэграванай у ёй камерай высокага дазволу, якая дазваляе ажыццяўляць распазнаванне вобразаў, сістэму фіксацыі адмоўным ціскам і спецыяльнага праграмнага забеспячэння для кантролю іх працы.

З мэтай кантролю ўсёй сістэмы аўтаматызаванай зборкі было распрацавана адмысловае праграмнае забеспячэнне для ПК, якое прадстаўляе з сябе Qt прыкладанне. Усё неабходнае апаратнае забеспячэнне падключаецца да ПК па інтэрфейсе USB. Кантроль над усімі элементамі сістэмы інтэграваны ў адно прыкладанне, названым PSAuto (ад аўтаматызаванай зборкі PS модуля). Дадатак напісана на мове праграмавання C++ і выкарыстоўвае Qt фреймворк версіі 4.8.7. Функцыянал прыкладання дазваляе ажыццяўляць поўны спектр працэдур аўтаматызаванай зборкі, яе кантролю і тэставанне ўсёй сістэмы.

Abstract

The degree work, 52 pages, 33 drawings (diagram, diagrams, graphs, histograms, photographs, drawings), 2 appendices, 12 sources.

AUTOMATION, SOFTWARE, QT DEVELOPMENT INSTRUMENT, OPENCV LIBRARY, MVC DESIGN TEMPLATE, IMAGE PROCESSING, TESTING, CMS PHASE II TRACKER UPGRADE.

The object of the study is an automated assembly system of CMS Phase II Tracker modules of the Large Hadron Collider (LHC), in particular, the software of this system.

The purpose of the work is to modernize the software for controlling the automated assembly of CMS Phase II Tracker modules of the LHC, upgrading the non-program elements of the automated assembly system, conducting system tests, and automated assembly of the very first prototype module.

The CMS phase II upgrade outer tracker is built from two types of modules (PS and 2S) each consisting of two silicon sensors and associated electronics and mechanics. In the case of the PS module, the sensors have to be assembled to a precision of approximately 40 μm . In order to satisfy this requirement and a short module assembly time, an automated assembly system is proposed. This system is based on a high-precision motion-stage integrated with high-resolution camera, which provides pattern-recognition, vacuum holding system and control systems.

In order to control the whole automated assembly system a PC-based Qt application was developed. All the necessary hardware is connected to the PC with USB interface. The control of the all parts the systems is integrated in a single software application henceforth referred to as PSAuto (from automated assembly of PS module). It is entirely written in C++ and utilizes the Qt framework version 4.8.7. The functionality of the application allows to perform a full range of automated assembly procedures, its control and testing of the entire system.

Используемые сокращения

БАК – Большой адронный коллайдер

СРСП – Сенсор-распорка-сенсор-пластина

AL-CF – Carbon-fibre reinforced aluminium

CF – Carbon-fibre

CMS – Compact muon solenoid

HL-LHC – High Luminosity LHC

LHC – Large Hadron Collider

PS – Pixel Strip (module)

2S – 2 Strip (module)

Введение

Большой адронный коллайдер (БАК) – крупнейший ускоритель частиц в мире. Во время его первого периода работы с 2010 по 2013 с его помощью были сделаны выдающиеся достижения. Возможно одно из наиболее известных – открытие теоретически предсказанного бозона Хиггса, за что Франсуа Энглер (François Englert) и Питер Хиггс (Peter Higgs) была присуждена Нобелевская премия по физике в 2013 году. Исследование явлений при столкновении частиц высоких энергий БАК происходит в четырёх позициях, расположенных вдоль основного кольца ускорителя и соответствующих четырём основным детекторам частиц – ATLAS, CMS, ALICE и LHCb [1].

Компактный мюонный соленоид

Компактный мюонный соленоид (Compact Muon Solenoid или CMS) – это цилиндрический детектор элементарных частиц, разработанный для измерения широкого спектра параметров различных элементарных частиц, производимых во время столкновений (коллизий) протонов и/или тяжёлых ионов, ускоренных в БАК. Детектор имеет длину около 28 метров и диаметр около 15 метров. Он является самым тяжёлым детектором элементарных частиц в мире и весит приблизительно 14000 тонн. Своё имя CMS детектор получил благодаря его 3-м ключевым характеристикам: его относительно *компактный размер*, его исключительная способность в детектировании и *измерении мюонов* и третья, его основная особенность, – *сверхпроводящий соленоид*, способный создать магнитное поле до 3.8 Т.

CMS детектор состоит из нескольких слоёв, каждый из которых выполняет свою отдельную роль в детектировании и измерении пролетающих сквозь них элементарных частиц. Схематическое изображение поперечного сечения данных слоёв и их задачи в восстановлении пути элементарной частицы показаны на Рисунке 1.

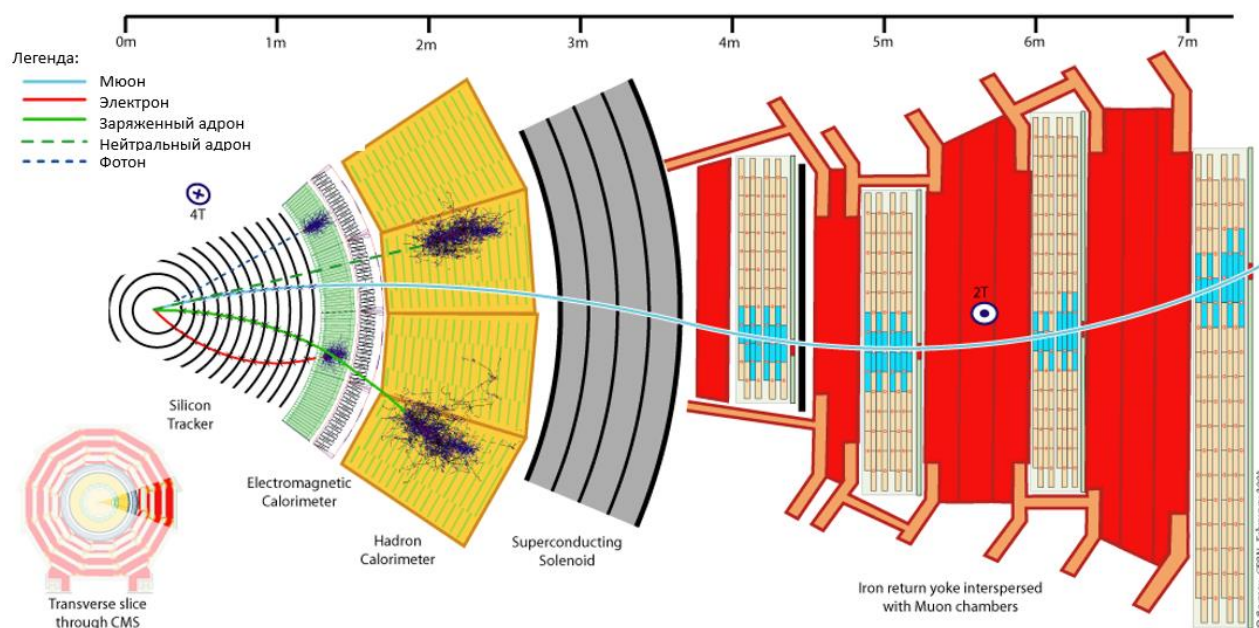


Рисунок 1 – Вид на слои CMS детектора в поперечном разрезе

Вторая стадия модернизации CMS трекера

После проведения второй стадии модернизации БАК будет обладать гораздо большей светимостью. Данный режим его работы получил название High Luminosity LHC (HL-LHC) (с англ. – БАК на большой светимости). Серьёзной сложностью условий работы данного режима БАК является намного больший поток производимых данных, который многократно превосходит пропускную способность электронных устройств ввода-вывода детекторов. Однако подавляющее большинство элементарных частиц, образованных в результате работы HL-LHC, малоинтересны в текущих исследованиях физики высоких энергий и характеризуются низким поперечным импульсом. Таким образом игнорируя попадания элементарных частиц с низким поперечным импульсом, можно значительно уменьшить объём считываемых данных. С целью обеспечения возможности распознавания момента импульса на уровне аппаратного обеспечения был создан проект двухслойных детектирующих модулей. Основная идея работы новых модулей состоит в разграничении проходящих частиц с низким и высоким поперечными импульсами, основываясь на приближенном измерении кривизны траектории, вызванной магнитным полем, в котором находится модуль. Например, частица с высоким поперечным импульсом после прохождения детектирующего пикселя/полоски первого сенсора пройдёт через соответствующий пиксель/полоску соседнего второго сенсора. В то время как частица с низким поперечным импульсом будет иметь большую кривизну траектории и пройдёт немного в стороне от соответствующего элемента первого сенсора на втором сенсоре. Варьируя расстояние между сенсорами и количество соседствующих детектирующих пикселей/полосок, необходимых для сопоставления областей прохождения на смежных сенсорах модуля (две соседних детектирующих полоски на Рисунке 2),

возможно выставить пороговое значение поперечного импульса для проходящей частицы [2].

Из Рисунка 2 видно, что частицы, которые пройдут в пределах двух соседних детектирующих полосок второго сенсора от соответствующий полоски первого сенсора, будут считаться как частицы с высоким поперечным импульсом.

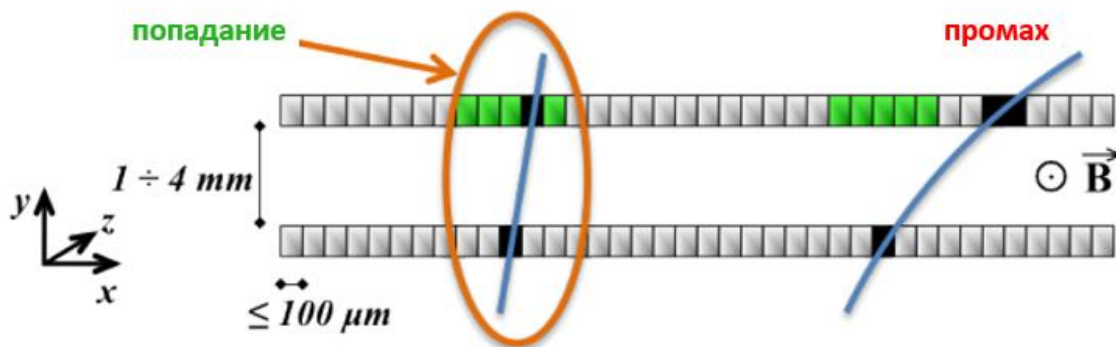


Рисунок 2 – Пример разграничения высокого и низкого поперечного импульса

Двухслойный детектирующий модули

Трекер CMS детектора второй стадии модернизации будет использовать два типа детектирующих модулей – 2S модули и PS модули. Чтобы достигнуть эффективного распознавания частиц с низким поперечным импульсом по всему объёму трека, в различных его частях сенсоры модулей будут располагаться на разном расстоянии друг от друга. Для 2S (PS) модулей расстояние между сенсорами будет варьироваться между 1.8 и 4 мм (1.6, 2.6 – 4 мм). PS модуль в разборном виде показан на Рисунке 3.

В PS модуле сенсоры приклеены к распоркам изготовленных из алюминия, усиленного углеродным волокном. Они обеспечивают необходимую теплопроводность, остро необходимую для охлаждения модуля. Два сенсора вместе с распорками, в свою очередь, приклеены к опорной пластине из углеродного волокна. Эта структура в дальнейшем будет называться сенсор-распорка-сенсор-опора (CPCO). Данная работа ориентирована исключительно

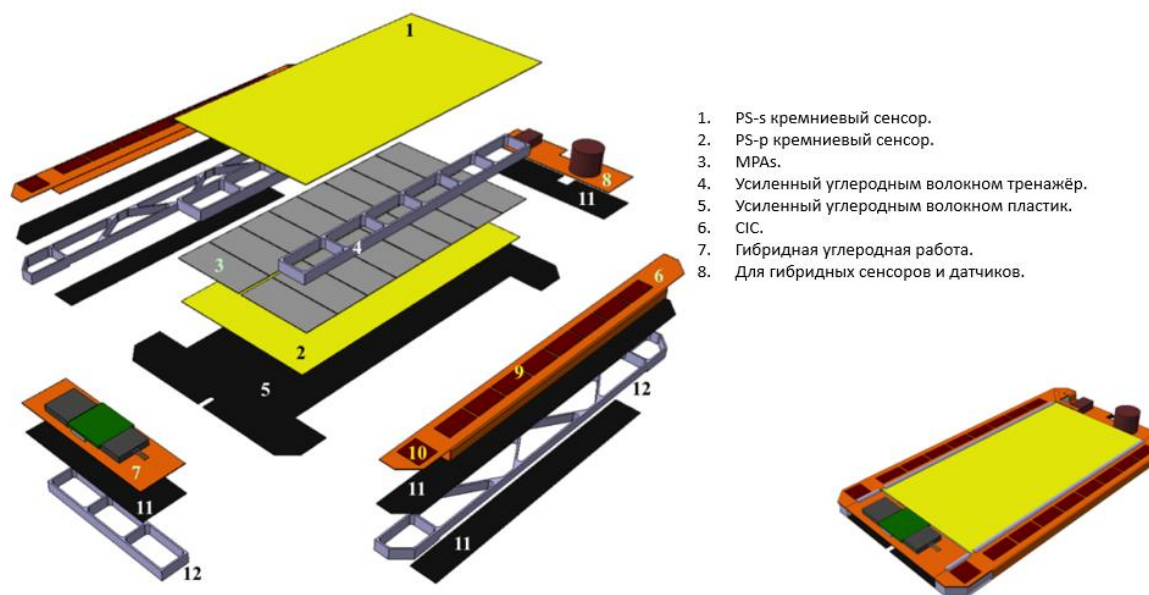


Рисунок 3 – PS модуль в разобранном виде

на сборку СРСО. Требуемые допущения по точности сборки СРСО показаны на Рисунке 4. Для PS модуля сенсоры должны быть расположены с относительной точностью в 40 мкм, измеряя с короткой стороны. Данная точность соответствует максимальной погрешности относительного вращения сенсоров на величину 0,8 мрад [3].

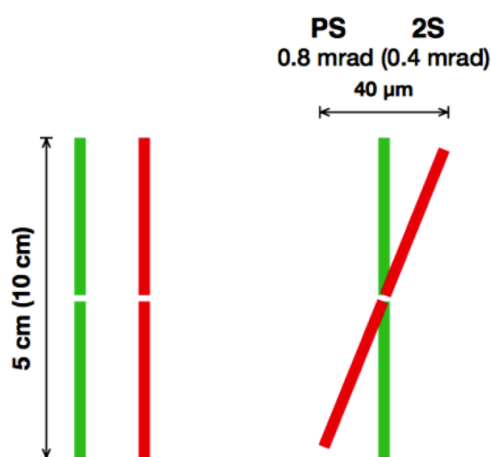


Рисунок 4 – Требования точности к сборке PS и 2S модулей

Глава 1

Сборка модулей

Сборку высокоточных детектирующих модулей такого рода обычно проводят вручную, используя изготовленные на заказ специальные механические фиксирующие платформы. В качестве альтернативы данному привычному методу предлагается использовать систему автоматизированной сборки.

1.1 Ручная сборка с помощью механической платформы

Один из способов сборки модулей – ручная сборка с помощью изготовленной на заказ специальной механической фиксирующей платформы, прототип которой показан на Рисунке 1.1 [4].



Рисунок 1.1 – Прототип механической фиксирующей платформы

При таком способе сборки все составные части размещаются на платформе и собираются вручную. Для обеспечения необходимой точности сборки фиксирующая платформа имеет три опорных стопора: один на длинной стороне и два вдоль более короткой стороны. С противоположных стопорам сторон собираемый модуль аккуратно прижимается специальными пружинами. Вместе они обеспечивают необходимую точность позиционирования компонентов модуля перед и во время сборки/склеивания модуля.

Однако, описанный выше способ сборки обладает рядом недостатков. В частности, для сборки с его помощью необходимо много времени и данная механическая система плохо масштабируется на промышленные масштабы сборки. Также данный метод обладает относительно плохой повторяемостью собранных модулей, пусть и в пределах необходимой точности, и не предоставляет возможностей контроля процесса сборки. Более того, фиксирующая платформа выставляет высокие требования к качеству её изготовления (точность выточки деталей должна составлять несколько микрон), а также стопоры требуют регулярной калибровки из-за возможных сдвигов и/или стачивание их поверхностей. Наконец, данная механическая система требует максимум участия человека в её работе. Несмотря на расчётную точность системы в теории, факт участия человека в сборке означает, что всегда будет существовать какой-то процент бракованных модулей, произведённых только по вине человека.

1.2 Система автоматизированной сборки

Предлагаемая система автоматизированной сборки состоит из трёх подсистем: двигательной подсистемы, подсистемы наблюдения и подсистемы отрицательного давления (Рисунок 1.2)

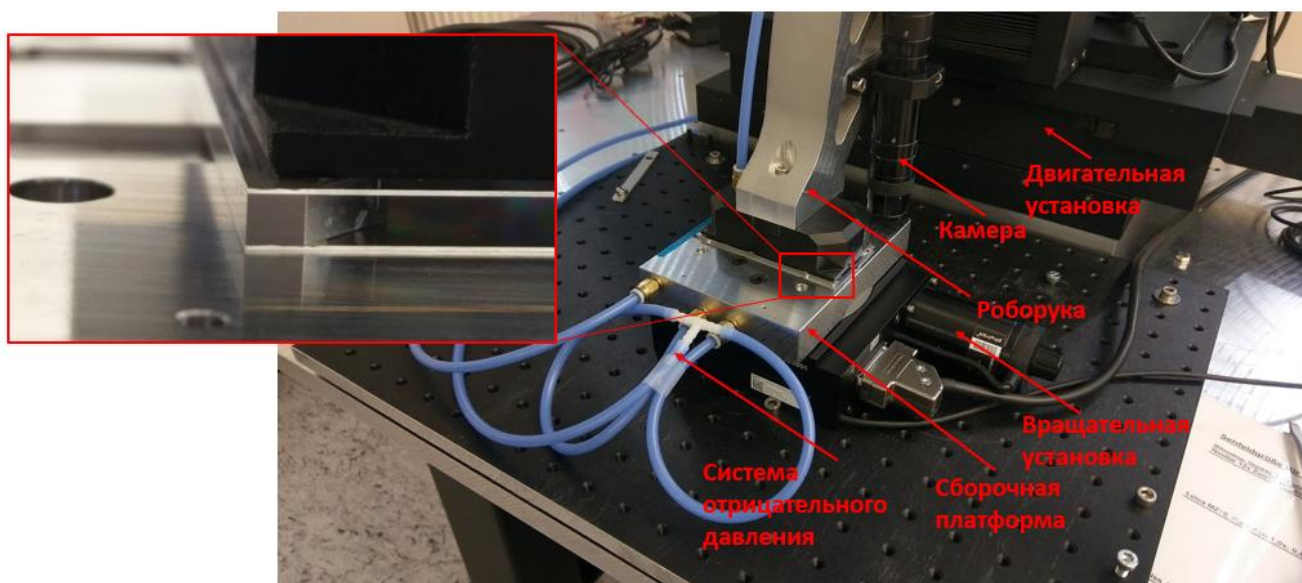


Рисунок 1.2 – Предлагаемая система автоматизированной сборки

Двигательная подсистема обеспечивает точные передвижения, необходимые для расположения компонентов, из которых состоит СРСП. Движение обеспечивается благодаря двум подвижным элементам двигательной подсистемы, обеспечивающим первый – по трём координатам, а второй – вращение. Специально изготовленное приспособление из алюминия, далее именуемое "роборукой", смонтировано на двигательной подсистеме так, что

обеспечивает движение по трём координатам, таким образом позволяя перемещать компоненты модуля в прямоугольных координатах. Сами компоненты модуля, в свою очередь, размещаются на вращательной установке, осуществляющей вращение в плоскости ХОУ той же прямоугольной системы координат. Элементы двигательной подсистемы управляются специальным блоком управления. Вся аппаратная часть двигательной подсистемы произведена фирмой Lang, обеспечиваемая точность перемещений которой не менее 4 мкм и 2 мрад по перемещению и вращению, соответственно.

Подсистема отрицательного давления позволяет закреплять компоненты собираемого модуля на роборуке и сборочной платформе. Она состоит из одного насоса, подводящей отрицательное давление к четырём переключаемым клапанам, которые, в свою очередь, распределяют отрицательное давление по четырём независимым линиям. Переключение клапанов осуществляется путём приложения управляющих сигналов напряжением в 12 В (12 В – включенное состояние, 0 В – выключенное). Управляющие сигналы обеспечиваются специальной релейной картой. Одна линия отрицательного давления подведена к устройству захвата компонентов модуля, крепящегося на конце роборуки, другие – к сборочной платформе. Захватное устройство сделано из ESD пластика, в котором располагается внутренняя камера для распределения подключённого отрицательного давления по специальным, направленным вниз, присоскам, которые слегка выступают за нижнюю грань захватного устройства. Схематичное изображение устройства захвата отрицательным давлением показано на Рисунке 1.3.

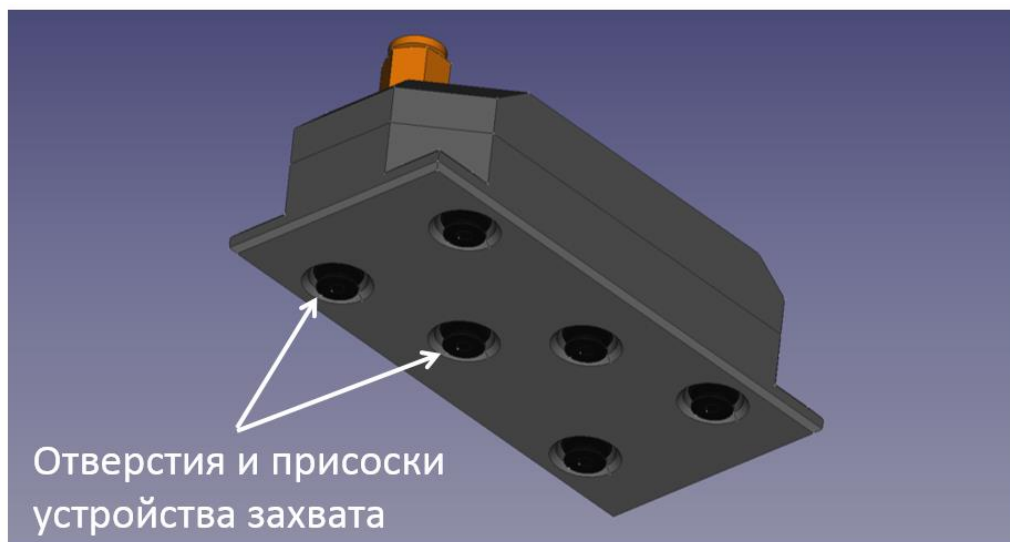


Рисунок 1.3 – Устройства захвата системы автоматизированной сборки (вид снизу)

Процедура захвата происходит путём касания присосками сенсора, включения подачи отрицательного давления устройству захвата и непосредственному перемещению роборуки вверх с захваченным сенсором. Процедура высвобождения сенсора начинается с контакта нижней поверхности прикрепленного сенсора (или опорной пластины) поверхности, на которую

необходимо поместить прикрепленный компонент. Далее выключается подача отрицательного давления и производится увод роборуки в вверх. С целью избегания нежелательных перемещений компонента модуля во время увода роборуки, на сборочной платформе компонент фиксируется специальными присосками снизу, принцип работы которых полностью схож с фиксацией на устройстве захвата.

Подсистема наблюдения делает снимки компонентов, проводит распознавание их расположения и ориентации в пространстве, что является особенно важным для осуществления точной сборки модуля. Она представлена камерой высокого разрешения фирмы IDS и системой линз, предоставляющих значительное оптическое увеличение снимаемого изображения. Данная структура крепится к роборуке и фиксируется в положении, направленном вниз. Основной задачей камеры является осуществление съёмки компонентов до и после их захвата и опускания вниз для последующей обработки этих снимков специализированным программным обеспечением, которое точно определяет положение компонента и его ориентацию в пространстве.

1.3 Сборочная платформа

Одной из наиболее важных частей системы автоматизированной сборки является сборочная платформа. Её основной задачей является фиксация модулей во время процедуры сборки модуля.

Сборочная платформа должна удовлетворять следующим требованиям:

1. Фиксировать все необходимые компоненты с помощью отрицательного давления.
2. Обеспечивать возможность расположения двух распорок модуля строго параллельно и на определённом расстоянии.
3. Быть по возможности лёгкой и иметь центр масс как можно ближе к оси вращения вращательной установки, на которой платформа будет установлена.

Чтобы выполнить упомянутые выше требования, был предложен следующий дизайн платформы (Рисунок 1.4).

Сборочная платформа имеет два углубления, в каждом из которых находятся по три опорных стопора обеспечивающих точное позиционирование распорок модуля. В сравнении с механической фиксирующей платформой, стопоры данной сборочной платформы являются частью платформы, а не прикручиваются отдельно, таким образом они не нуждаются в регулярной калибровке. Однако, наличие данных углублений делает невозможным размещение других компонентов модуля (сенсоров и опорных пластин) непосредственно поверх этих углублений, так как слишком большая часть их

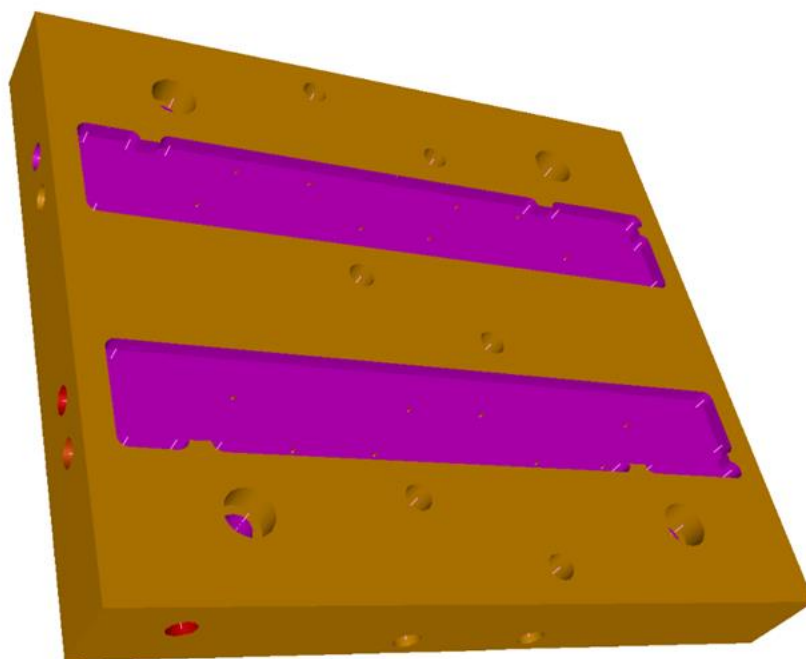


Рисунок 1.4 – Эскиз сборочной платформы

площади остаётся без поддержки под давлением устройства захвата сверху во время склеивания компонентов модуля. Данная проблема была решена путём размещения плоских компонентов модуля (сенсоры и опорные пластины) перпендикулярно распоркам на платформе. В данном случае только небольшая часть данных компонентов не будет иметь поддержки снизу, что не является критичным. Перпендикулярное размещение некоторых компонентов собираемого модуля может быть с лёгкостью осуществлено вращательной платформой, на которой находится сборочная платформа. Центр масс спроектированной платформы находится очень близко к оси вращения – меньше 1 мм. В то время как её масса составляет около 1 кг, таким образом никаких нежелательных эффектов, влияющих на точность работы вращательной платформы, не ожидается.

Внутри сборочной платформы располагаются две независимые линии отрицательного давления: первая – для удержания распорок, вторая – для удержания сенсоров и опорных пластин. Линия распорок распределяет давление по ряду отверстий (0,7 мм в диаметре) на дне углублений для распорок. Размер и расположение данных отверстий определяется формой распорки (Рисунок 1.5). Эти отверстия не оборудованы присосками из-за своего очень малого размера. Присосок таких размеров попросту нет на рынке. Однако, соприкасающиеся поверхности (дно углублений платформы и соответствующая грань распорки) имеют достаточную плоскость, чтобы утечка отрицательного давления была в допустимых пределах и распорки были зафиксированы достаточно плотно.

Точки приложения отрицательного давления сборочной платформой

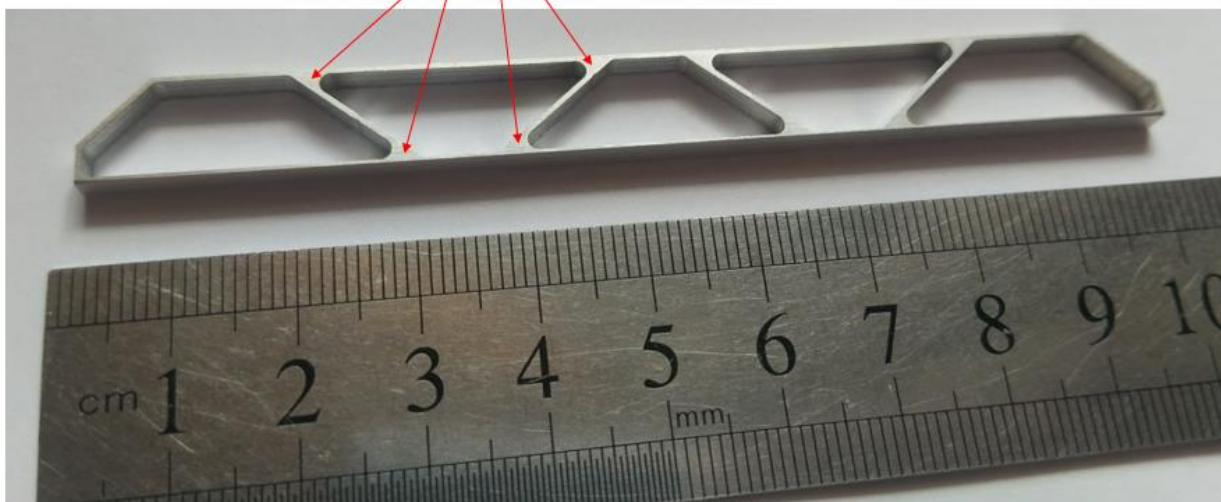


Рисунок 1.5 – Алюминиевый прототип распорки для PS модуля

Вторая линия отрицательного давления распределяет его между присосками, удерживающими сенсоры или опорные пластины. В сборочной платформе используются присоски, абсолютно идентичные захватывающему устройству. Так же, как и в захватывающем устройстве, они слегка выступают за поверхность сборочной платформы. Однако, они не должны препятствовать процессу сборки, а именно этапу склеивания сенсора с распорками (этапы сборки будут подробно описаны в параграфе 1.5). Именно поэтому высота углублений меньше толщины распорок, что позволяет верхней части распорок быть выше верхней части присосок. Другими словами, присоски не препятствуют этапам сборки, в которых не участвуют, так как находятся ниже верхней части распорок.

1.4 Быстросохнущее клеящее вещество

Другой важной частью системы автоматической сборки является клеящее вещество, используемое во время сборки, говоря точнее – время его высыхания, толщина слоя и прочность соединения. Другими словами, мало смысла оставлять один модуль на длительное время зафиксированным в установке, просто ожидая пока высохнет клеящее вещество. Например, основное клеящее вещество, используемое в сборке PS модуля, требует 24 часа на высыхание. Именно поэтому система автоматизированной сборки нуждается в технологии, которая позволит избежать длительного ожидания. Одно из предложенных решений – использование небольшого количества быстросохнущего клеящего вещества в дополнении к основному. Данное вещество должно удовлетворять следующим требованиям:

1. Обеспечивать необходимую прочность соединения спустя 15 минут применения.
2. Не вступать во взаимодействие с основным клеящим веществом.
3. Обеспечивать тонкий слой (до 30 мкм).

Более того, упомянутые выше требования должны быть выполнены с использованием как можно меньшего количества вещества. Для тестирования клеящих веществ использовались небольшие алюминиевые бруски формы прямоугольного параллелепипеда (в качестве приближения распорок из алюминия, усиленного карбоновым волокном) и образцы стела (в качестве приближения кремневым сенсорам). Достижение перечисленных требований в высокой степени зависит от двух факторов: первый – способ применения клеящего вещества, второй – непосредственные свойства этого вещества.

Возможны множество способов применения быстросохнущего клеящего вещества, среди них: несколько его капель внутри слоя основного клеящего вещества, несколько его капель по краям соединения, заполнение скоса распорок быстросохнущим клеящим веществом, несколько его капель поверх края соединения и др. Для описываемой системы было принято решение проверить заполнение скоса стопоров быстросохнущим клеящим веществом и несколько его капель по краям соединения. Несмотря на то, что проклеивание по скосу может обеспечить необходимую прочность соединения после 15 минут, намного практичнее применять несколько капель вещества по краям соединения. Оно применялось способом, показанным на Рисунке 1.6. На практике достаточно капель намного меньших размеров, чем на Рисунке 1.6, для обеспечения необходимой прочности соединения.



Рисунок 1.6 – Способ применение быстросохнущего вещества во время тестов

Вторая часть вопроса – поиск подходящего быстросохнущего вещества на рынке. Основная сложность вопроса состоит в комбинации двух свойств:

быстрое высыхание и низкая вязкость. Низкая вязкость есть неотъемлемое свойство клеящего вещества, если необходимо обеспечить тонкий его слой. Однако, быстрое высыхание означает, что вещество должно стать твёрдым в короткое время, чего в свою очередь намного проще достичь, если вещество изначально будет большей вязкости. Несмотря на данные сложности, были подобраны несколько вариантов для тестирования:

1. *Polytec EP 660.* Согласно технической документации [5], время его высыхания составляет до 16 часов, что выходит далеко за рамки требований. Однако, производитель отметил, что необходимая прочность соединения может быть достигнута без полного высыхания. К сожалению, данное вещество, ожидаемо, не обеспечило необходимую прочность соединения по прошествии 15 минут. Единственным его преимуществом являлся факт того, что основное клеящее вещество изготавливается тем же производителем.
2. *Loxreal 31-42.* Согласно технической документации [6], время его полного высыхания составляет $\approx 20 - 30$ минут, в то время как время "схватывания" составляет $\approx 3 - 8$ минут. В результате после 15 минут данное вещество обеспечивает необходимую прочность соединения.
3. *Wekem WK5.* Согласно технической документации [7], время его высыхания составляет около пяти минут. Данный образец также обеспечивает необходимое качество соединения после 15 минут высыхания, но клеящее вещество от Loxreal имеет лучшее качество и удобнее в применении. Более того, Loxreal 31-42 обеспечивает более тонкий слой при высыхании под одинаковым давлением – ≈ 20 мкм – в то время как Wekem – ≈ 40 мкм (Рисунок 1.7).

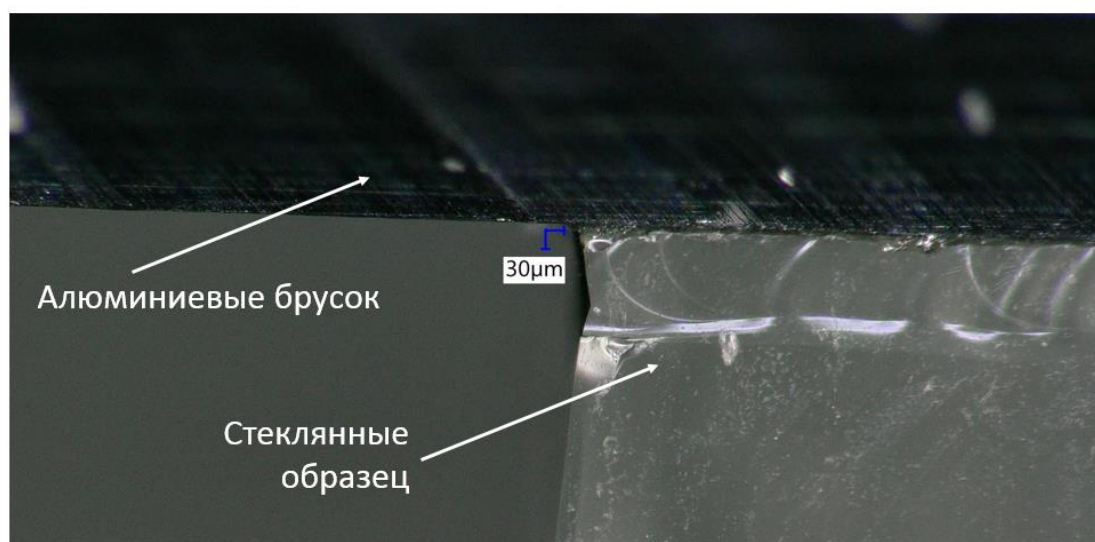


Рисунок 1.7 – Толщина слоя быстросохнущего клеящего вещества Loxreal 31-42 при высыхании под давлением ≈ 20 г/см²

Подводя итоги, Loheal 31-42 показал лучшие результаты, удовлетворив все требования: он обеспечивает необходимую прочность соединения после 15 минут высыхания под небольшим давлением ($\approx 20 \text{ г/см}^2$), создавая при этом тонкий слой – $\approx 20 \text{ мкм}$, и демонстрирует отсутствие взаимодействия с основным клеящим веществом.

1.5 Процесс автоматизированной сборки

Процесс автоматизированной сборки СРСП для PS модуля состоит из следующих шагов:

1. *Подготовка верхнего сенсора.* Сперва необходимо поместить верхний сенсор на сборочную платформу и зафиксировать отрицательным давлением. После – распознать его позицию и ориентацию в пространстве с помощью камеры и специального программного обеспечения и сохранить данную информацию. Затем захватить и поднять его с помощью устройства захвата и оставить прикреплённым. Начиная с самого первого шага подача отрицательного давления на устройство захвата не прекращается в течении всего процесса сборки.
2. *Подготовка распорок.* Как только верхний сенсор прикреплен к устройству захвата и поднят с платформы, сборочная платформа должна повернуться на 90 градусов таким образом, что распорки будут ложиться в специально отведённый для них углубления корректно ориентированными относительно верхнего сенсора. Далее их необходимо аккуратно прислонить к опорным стопорам и включить подачу отрицательного давления для их фиксации. После система должна найти и определить положение опорного маркера на сборочной платформе, что позволит однозначно рассчитать положение распорок в пространстве. Отрегулировать их ориентацию с помощью вращательной установки таким образом, чтобы они были параллельны оси ОХ двигательной установки, что означает быть сонаправленным верхнему сенсору.
3. *Склеивание верхнего сенсора с распорками.* Как только верхний сенсор и распорки готовы, их положения и ориентации определены, программное обеспечение может вычислить путь для устройства захвата с удерживаемым им верхним сенсором к распоркам. После необходимо поместить основное клеящее вещество на распорки и несколько капель быстросохнущего вещества на краях распорок и передвинуть устройство захвата в расчётную позицию для склеивания с распорками. Особое внимание стоит уделить движению устройства захвата по координате Z, так как это напрямую влияет на толщину слоя клеящего вещества.

4. *Подготовить нижний сенсор.* По истечении 15 минут капли быстросохнущего вещества обеспечивают необходимое соединение для извлечения склеенных распорок и верхнего сенсора. Для этого сделать необходимо сначала отключить подачу отрицательного давления, удерживающего распорки, и после просто поднять устройство захвата, которое уже будет удерживать не только верхний сенсор, но и приклеенные к нему распорки. После того, как платформа освобождена, её следует повернуть на 90 градусов так, что нижний сенсор может быть помещён на платформу и может быть зафиксирован отрицательным давлением. Далее необходимо так же, как и для верхнего сенсора, определить его положение и ориентацию с помощью камеры и специального программного обеспечения и сохранить данную информацию. Отрегулировать ориентацию сенсора с помощью вращательной установки таким образом, чтобы он был параллелен оси ОХ двигательной установки, что означает быть сонаправленным верхнему сенсору и распоркам.
5. *Склеивание нижнего сенсора со структурой «распорки + верхний сенсор».* Все необходимые для данного этапа компоненты зафиксированы и их расположение известно. Следовательно, программное обеспечение способно рассчитать путь для устройства захвата и переместиться в позицию склеивания после нанесения главного и быстросохнущего клеящего вещества.
6. *Подготовка опорной пластины.* По прошествии 15 минут склеенная структура может быть поднята. Далее необходимо поместить опорную пластину на сборочную платформу согласно трём опорным штифтам и отверстиям для них в самой пластине и зафиксировать её с помощью отрицательного давления. Данные штифты не представлены в текущей версии сборочной платформы.
7. *Склеивание структуры «сенсор-распорки-сенсор» с опорной пластиной.* Так как программное обеспечение уже знает расположения опорного маркера на сборочной платформе, следовательно, оно способно вычислить позицию склеивания для текущего этапа. Далее устройство захвата передвигается в эту позицию после нанесения основного и быстросохнущего клеящего вещества.
8. *Автоматическая сборка завершена.* По прошествии 15 минут быстросохнущее вещество обеспечивает необходимую прочность соединения и собранный СРСП может быть убран со сборочной платформы и оставлен в другом месте на 24 часа до полного высыхания склеивающих веществ. Однако опорные штифты опорной пластины очень плотно подходят самой пластине, следовательно, ручное снятие СРСП может с высокой долей вероятности вызвать его разрушение. Поэтому лучше доверить и данный этап двигательной установке, которая может безопасно поднять собранный СРСП вертикально вверх, сняв его с опорных штифтов.

Глава 2

Программное обеспечение контроля сборки

С целью контроля всей системы автоматизированной сборки было разработано специальное программное обеспечение (Qt приложение) для ПК. Всё необходимое аппаратное обеспечение подключается по интерфейсу USB.

2.1 Общая структура приложения

Контроль подсистемами движения, наблюдения и отрицательного давления интегрированы в одно приложение, именуемого в дальнейшем PSAuto (от автоматизированной сборки PS модуля). Приложение написано на языке программирования C++ и использует Qt фреймворк версии 4.8.7. Схематически интеграция трёх подсистем автоматизированной сборки изображена на Рисунке 2.1.

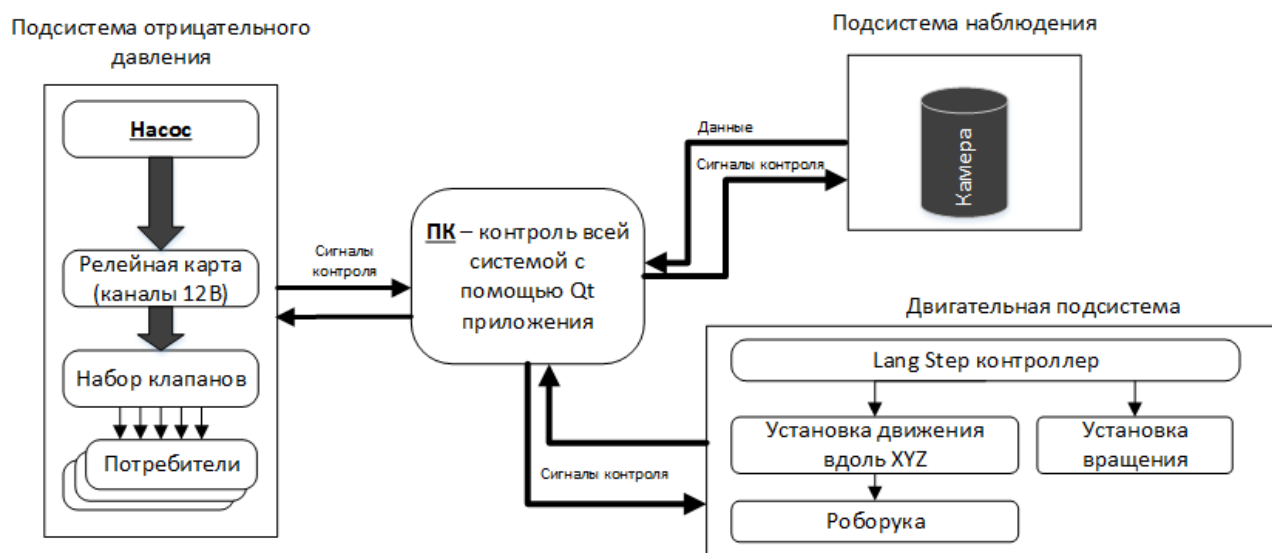


Рисунок 2.1 – Схематическое изображение интеграции подсистемами движения, наблюдения и отрицательного давления с помощью Qt приложения PSAuto

2.1.1 Шаблон проектирования «Модель-Представление-Контроллер»

Структура приложения построено согласно шаблону проектирования «Модель-Представление-Контроллер» (Model-View-Controller (MVC)). В нём рассматриваются три основных типа объектов: объекты *Модели*, объекты *Представления* и объекты *Контроллера*. При разработке приложения важным шагом является выбор или создание пользовательских классов для объектов,

которые попадают в одну из этих трех групп. Каждый из трех типов объектов отделен от других абстрактными границами. Шаблон определяет не только роли объектов в приложении, но также и способ взаимодействия объектов друг с другом [8]. Ключевым моментом MVC является то, что объекты Представления и Контроллера зависят от объектов Модели, однако объекты Модели в свою очередь не зависят от них. Взаимодействие данных типов объектов схематически показано на Рисунке 3.2.

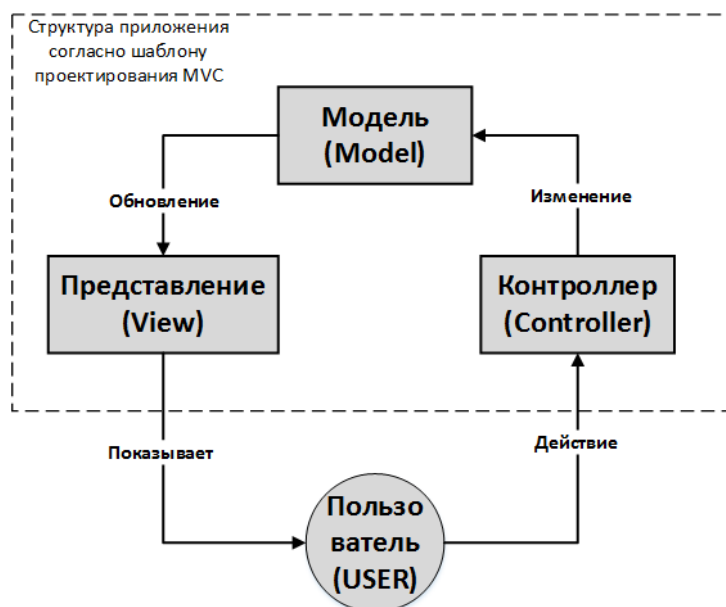


Рисунок 2.2 Модель, Представление и Контроллер (MVC) относительно пользователя (классическое взаимодействие)

На Рисунке 2.2 показано взаимодействие объектов классической архитектуры MVC. Существует различные варианты данной диаграммы с точки зрения взаимодействия между объектами MVC. Это зависит от типа приложения и его реализации. Например, в приложении PSAuto данная диаграмма будет иметь вид как на Рисунке 2.3.

По сравнению с классической структурой MVC, в данном случае Модель напрямую не информирует/обновляет Представление. Эта информация проходит через Контроллер, который действует как посредник между Моделью и Представлением. Контроллер часто отвечает за то, чтобы Представление имело доступ к объектам Модели, которые им нужно отображать, и выступает в качестве канала, через который объект Представления узнаёт об изменениях в Модели. Объекты Контроллера также могут выполнять задачи настройки приложения и управлять рабочими процессами других объектов [8].

В зависимости от логики и требований приложения, Модель, Представление и Контроллер следуют определённым требованиям и обладают определёнными свойствами.

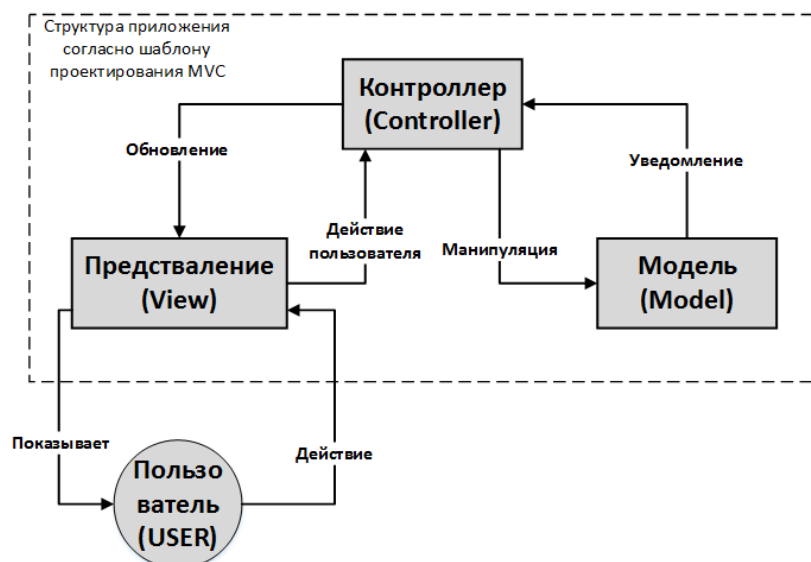


Рисунок 2.3 – Модель, Представление и Контроллер (MVC) относительно пользователя (взаимодействие в приложении PSAuto)

Модель является центральным компонентом системы. Она напрямую работает с данными, определяя логику и правила работы приложения. Более того, как в случае с PSAuto, Модель ещё и регулирует взаимодействие между программным и аппаратным обеспечением. Например, в приложении PSAuto есть объект Модели *ConradModel*, который отвечает за взаимодействие с релейной картой, контролирующей линии отрицательного давления системы. Данный объект полностью согласуется с шаблоном проектирования MVC как объект Модели, так как не зависит от других классов, однако другие классы зависят от него, используя его возможности для своего функционирования.

Представление объединяет все объекты, отвечающие за графический интерфейс приложения: окна, вкладки, формы, поля, кнопки и т.п. Объекты представления отвечают за отображение информации, предоставляемой объектами модели, а также предоставляет пользователю вносить изменения в работы приложения (осуществлять контроль процесса сборки). Например, в приложении PSAuto одна из задач объекта *AssemblyModuleAssembler* контролирует отображение информации о состоянии линий отрицательного давления в реальном времени, а также элементы их управления. Однако, он только отображает элементы контроля и передаёт действия пользователя объектам Контроллера, но сам не участвует в обработке этих действий.

Контроллер в основном ответственный за приём и обработку поступающих данных, генерацию команд для Модели и Представления. Например, в приложении PSAuto объект *ConradManager* предоставляет весь необходимый функционал для контроля линиями отрицательного давления и получения информации об их текущем статусе. Также стоит отметить, что очень часто не существует прямой связи между Моделью и Представлением, как показано на Рисунке 2.2. Вместо этого Контроллер выступает в качестве промежуточного звена, передающего информацию между Моделью и Представлением.

2.1.2 Библиотека OpenCV

Множество алгоритмов, обеспечивающий функционал приложения PSAuto, реализуется при помощи библиотеки с открытым кодом – OpenCV (*Open Source Computer Vision*). Это кроссплатформенная библиотека алгоритмов, в основном ориентированных на компьютерное зрение и обработку изображений в реальном времени (OpenCV – от англ. Computer Vision). Данная библиотека была разработана в исследовательском центре Intel в Нижнем Новгороде (Россия). Она распространяется на условиях лицензии BSD и может быть свободно использована в академических целях [9].

OpenCV была специально ориентирована на высокую вычислительную эффективность и, как уже упоминалось, ориентированность на работу в реальном времени. Она полностью реализована на языке программирования C++ и способна использовать преимущества многоядерных процессорных систем. Также она поддерживает возможность автоматической оптимизации под архитектуру продуктов Intel с помощью библиотек *Integrated Performance Primitives (IPP)*, которые состоят из низкоуровневых оптимизированных подпрограмм в различных алгоритмических областях [10].

Одна из основных задач библиотеки OpenCV – предоставить простую и понятную в использовании инфраструктуру для работы с приложениями компьютерного зрения, что оказывает серьёзную поддержку в разработке достаточно сложных приложений данной области. Библиотека насчитывает более 500 функций, охватывающих множество областей компьютерного зрения и обработки изображений. Ниже приведены основные функции библиотеки OpenCV:

- Ввод/вывод изображений и видео (открытие изображений из файла или непосредственное получение от камеры, вывод изображений/видеофайлов)
- Операции линейной алгебры над векторами и матрицами (умножение, определители, собственные значения, сингулярное разложение).
- Различные динамические структуры данных (списки, очереди, наборы, деревья, графики).
- Основы обработки изображений (фильтрация, обнаружение краев, определение угла, выборка и интерполяция, преобразование цвета, морфологические операции, гистограммы и т.д.).
- Структурный анализ (связанные компоненты, обработка контуров, дистанционное преобразование, совпадение шаблонов, преобразование Хафа, многоугольное приближение, линейная подгонка, эллиптическая подгонка, триангуляция Делоне).
- Калибровка камеры (поиск и отслеживание шаблонов калибровки, калибровка, оценка фундаментальной матрицы, оценка гомографии, стереозапись).

- Анализ движения (оптический поток, сегментация движения, отслеживание).
- Распознавание объектов (метод собственных значений, метод скрытой Марковской Модели).
- Основные элементы графического интерфейса (отображение изображения/видео, управление клавиатурой и мышью, полосы прокрутки).
- Маркировка изображений (линии и другие фигуры, текстовые подписи).

Применение функционала библиотеки будет описано в следующих параграфах.

2.2 Распознавание образов

Распознавание образов является важной функцией всей системы автоматизированной сборки, в частности – подсистемы наблюдения. Она обеспечивает программное обеспечение информацией о *расположении* платформы и компонентов модуля в пространстве и их *планарной ориентации*. С помощью этой информации программное обеспечение способно вычислить, куда перемещать каждый компонент модуля во время сборки. По определению, планарная ориентация – это вращательная ориентация объекта в горизонтальной плоскости (ХОУ в данном случае) [3].

Для системы автоматизированной сборки планарная ориентация проводится в два этапа: независимое определение *положения* и *ориентации* каждого из четырёх маркеров на углах сенсора и совокупность положений четырёх маркеров, из которой вычисляется конечная точная ориентация всего сенсора. Это делается путем обработки снятых изображений *точных маркеров* в углах сенсоров (Рисунок 2.4), полученных подсистемой наблюдения. Маркеры точно расположены относительно полосок/пикселей сенсора, следовательно, узнав точное положение маркеров, точное позиционирование полосок/пикселей гарантировано, что и требуется для выполнения идеи двухслойных детектирующих модулей CMS трекера второй стадии модернизации. В качестве входных данных алгоритм берет необработанные изображения и возвращает значения положения и ориентации маркеров, расположенных в углах сенсора PS модуля. Алгоритм распознавания образов использует библиотеку *OpenCV*. Ниже перечислены этапы распознавание положения и ориентации сенсора:

1. Предварительная обработка изображения, полученного с камеры.
2. Определение позиции и планарной ориентации маркера.

3. Определение приблизительного положения следующего маркера и/или конечное вычисление планарной ориентации сенсора.

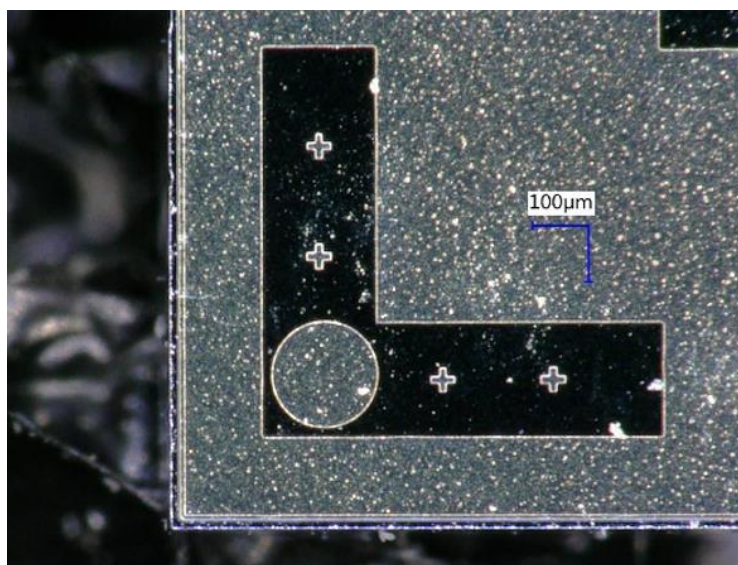


Рисунок 2.4 – Фотография маркера на углу фиктивного сенсора PS модуля

2.2.1 Предварительная обработка изображения, полученного с камеры

Необработанное изображение с камеры сперва преобразуется из цветного в изображение в оттенках серого, что известно, как *"grayscale"* (от англ. grayscale – оттенки серого) в библиотеке OpenCV. Пиксели изображения в оттенках серого содержат информацию об интенсивности в виде одного числа от 0 до 255, характеризующего степень темноты оттенка серого, в отличие от информации об интенсивности и цвете, содержащейся в пикселях цветного изображения. Поскольку маркеры основаны не на цветах, а на простых формах (окружность, угол, кресты), то информация о цвете не является полезной и потому не учитывается. Вместо этого изображение в оттенках серого преобразуется в *бинарное* изображение, у которого каждый пиксель либо белый, либо чёрный. Данная процедура получила название *"Thresholding"*, от англ. threshold – порог (пороговое значение). Операция Thresholding заключается в простом конвертировании каждого пикселя изображения в оттенках серого в белый (чёрный) пиксель, если его интенсивность выше(ниже) заданного порогового значения. Данная операция призвана уменьшить различие между изображениями маркеров идентичных сенсоров из-за случайного *шума*, вызванного пылью и случайными различиями на поверхностях маркеров сенсора. Примеры изображения маркера в оттенках серого и его бинарных изображений для различных пороговых значений показаны на Рисунке 2.5. Оптимальное значение порогового значения зависит от общего уровня освещённости вокруг места проведения сборки и контраста маркера на

поверхности сенсора. На данный момент, типичное значение порогового значения составляет около 90, что приблизительно соответствует освещению лаборатории лампами дневного света.

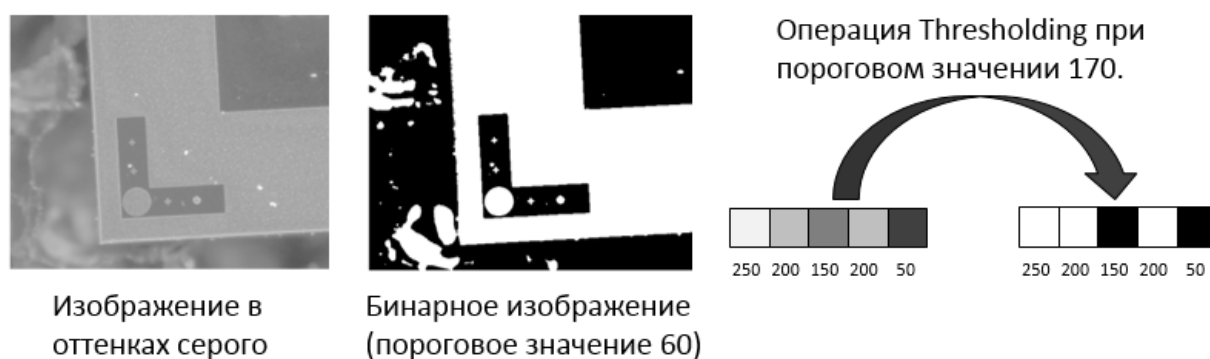


Рисунок 2.5 – Применение операции *Thresholding* на изображении маркера

2.2.2 Определение позиции и планарной ориентации маркера

Положение и ориентация маркера на бинарном изображении определяется с использованием стандартного для такой ситуации метода обработки изображений, известным как *сопоставление шаблонов*. Библиотека OpenCV предоставляет одноимённую функцию для выполнения таких задач. Метод находит ту часть исследуемого изображения, которая больше всего походит на шаблонное изображение. В случае автоматизированной сборки исследуемым изображением является преобразованное в бинарное изображение снимка маркера, полученного с камеры, а шаблонным – заранее подготовленное бинарное изображение маркера.

Сопоставление шаблонов начинается с итеративного сопоставления шаблонного изображения в каждой точке исследуемого изображения и вычисления определённого числового значения, или метрики, показывающего степень схожести шаблонного изображения с частью исследуемого изображения, которая соответствует данной точке. В приложении PSAuto эта точка является левой-верхней точкой данной части исследуемого изображения. Библиотека OpenCV предоставляет множество вариантов вычисления значения этой метрики. Аналогичные результаты наблюдаются для каждой из возможных метрик, поэтому была выбрана наиболее скоростная по вычислительной сложности, метрика, основанная на нормированной квадратичной разности интенсивностей налагающихся пикселей исследуемого изображения и шаблона:

$$R(x, y) = \frac{\sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2}{\sum_{x', y'} \sqrt{\sum_{x', y'} T(x', y')^2 \cdot \sum_{x', y'} I(x + x', y + y')^2}}$$

где I означает *исследуемое изображение*, T – *шаблонное изображение* и R – *результатирующая метрика*.

Данную метрику можно найти под именем CV_TM_SQDIFF_NORMED в библиотеке OpenCV. Точка исследуемого изображения, в которой метрика достигнет минимального значения показывает наиболее вероятное положение шаблонного изображения, маркера в случае автоматизированной сборки. На Рисунке 2.6 изображен процесс определения положения маркера с помощью сопоставления шаблона и его результат с использованием маркера на пробном сенсоре. После нахождения наиболее вероятного положения маркера алгоритм выделяет его белым прямоугольником на исследуемом изображении. Наблюдаемое на Рисунке 2.6 местоположение близко соответствует ожиданиям.



Рисунок 2.6 – Методика сопоставления шаблонов проиллюстрирована на левом изображении

На Рисунке 2.6 красные стрелки указывают направления итеративного расчета метрики в каждой точке основного изображения. Результат работы алгоритма сопоставления шаблонов при использовании тестовых изображений показан на правом изображении. Наиболее вероятное расположение маркера в главном изображении обозначено красным прямоугольником на Рисунке 2.6.

Чтобы вычислить ориентацию маркера в плоскости, перпендикулярной оптической оси камеры, процедура сопоставления шаблонов итеративно повторяется с различными вращательными преобразованиями для разных углов (в установленных пределах с определённым шагом), применяемыми к части исследуемого изображения, где предварительно маркер уже был найден без информации о его ориентации на плоскости. В каждой итерации сохраняется минимальное значение метрики, а по окончании цикла строится график

зависимости значения метрики от угла поворота исследуемого изображения. В силу симметричности формы маркера, данный график напоминает параболу, ветви которой направлены вверх, с минимум в точки того угла поворота исследуемого изображения, при котором шаблон лучше всего накладывается на исследуемое изображение. На Рисунке 2.7 схематично изображено определение планарной ориентации маркера, в правой части которого изображён график, соответствующий тесту, выполненному с изображением сенсора, где он имел планарную ориентацию около 3,5 градуса. Соответственно на графике чётко виден минимум при $\approx 3,5$ градусах. Большая точность измерение ориентации сенсора может быть достигнута путём оптимизации таких факторов, как окружающее освещение, фокусировка изображения и форма маркера. Наилучшая точность, достигнутая во время тестов, составила 0,025 градуса, что более чем достаточно для целей проекта, так значение ориентации, полученное таким способом, используется лишь для перепроверки конечной ориентации сенсора.

На левом изображении Рисунка 2.7 схематично изображена оценка планарной ориентации сенсора. Красные стрелки указывают направление вращательного преобразования, применяемого к основному изображению. График зависимости минимального значения метрики от углового преобразования показан на правом изображении. Наблюдается явный минимум при $\approx 3,5$ градусах.

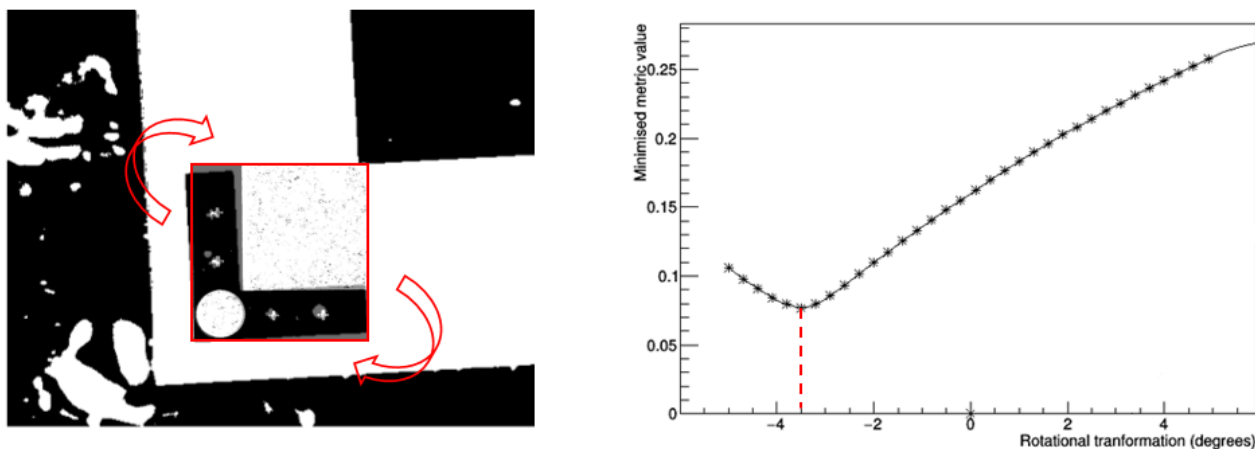


Рисунок 2.7 – Схематичное изображение определение планарной ориентации маркера

2.2.3 Определение приблизительного положения следующего маркера и/или конечное вычисление планарной ориентации сенсора

Процедура, описанная на шаге 2, повторяется для каждого угла сенсора. Значение планарной ориентации, определенные в каждом углу, используются

для того, чтобы задать направление движения, необходимое для того, чтобы роборука автоматически переместилась к соседнему углу. Конечная позиция и ориентация датчика определяются с помощью критерия согласия для четырех точек (x, y), соответствующих четырём маркерам в углах каждого сенсора. Полученное таким образом значение ориентации сенсора сверяется с приближёнными значениями ориентации для маркера каждого из четырёх углов. Если они совпадают в пределах допустимой погрешности, то это значение ориентации сенсора может использоваться в дальнейшем.

Подробная UML диаграмма алгоритма распознавания образов в системе автоматизированной сборки продемонстрирована в Приложении А. В Приложении Б продемонстрирован исходный код метода приложения PSAuto, отвечающего за процедуру распознавания образов.

2.3 Функционал приложения

Главное окно приложения PSAuto имеет несколько кнопок и флагов в верхней части и ряд вкладок ниже, включая следующие: Finder, Threshold, Assembly, Autofocus, Motion Manager и другие.

Finder (от англ. finder – искатель). Простая вкладка, единственной задачей которой является демонстрация последнего снимка, снятого камерой. Также в этой вкладке это изображение можно сохранить.

Threshold (от англ. threshold – порог, граница). Результат операции Thresholding в высокой степени зависит от условий освещения, поэтому каждый

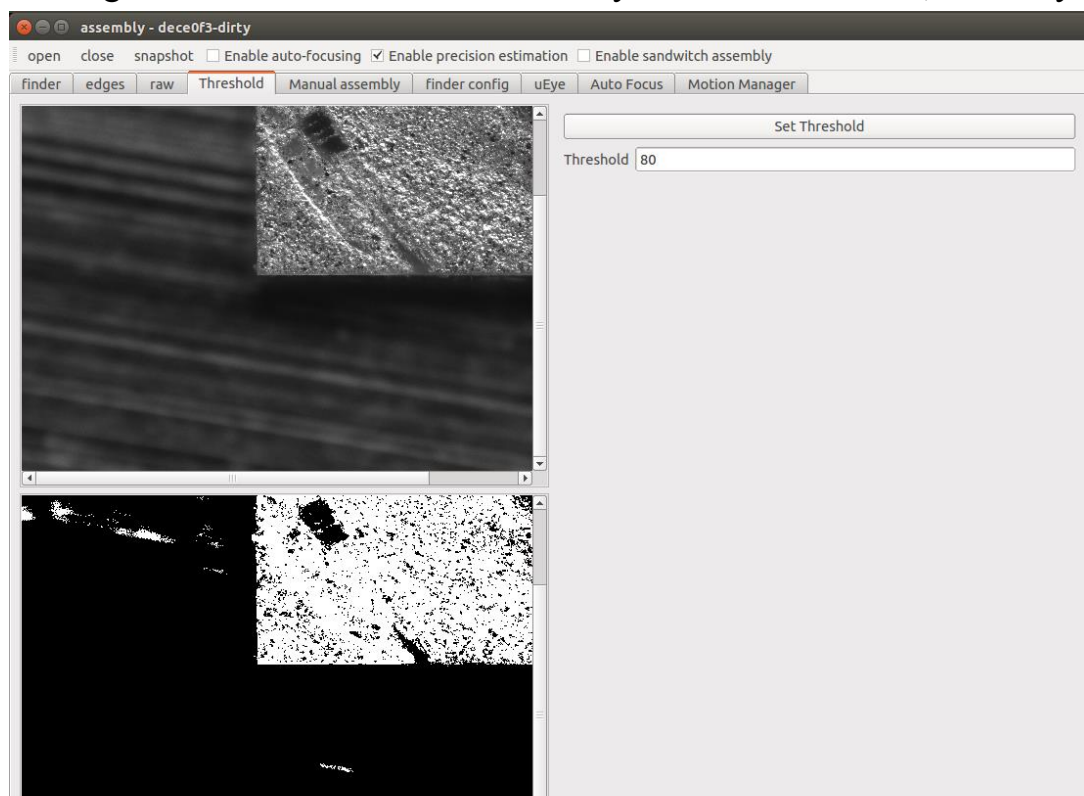


Рисунок 2.8 – Вид вкладки Threshold приложения PSAuto

новый тест требует калибровки значения применяемого порогового значения для максимально эффективной работы операции Thresholding. Данная вкладка специально создана для осуществления контроля над данным значением. С её помощью можно настраивать пороговое значение и получать мгновенный отклик в виде применения операции Thresholding к последнему снятому изображению с новым пороговым значением. Изображения до и после применения данной операции также демонстрируются в левой части вкладки. Вид вкладки Threshold показан на Рисунке 2.8.

Assembly (от англ. assembly – сборка). Это основная вкладка приложения (Рисунок 2.9). Она включает в себя четыре окна с изображениями слева, средства управления справа и статус двигательной установки в реальном времени. Левое-верхнее окно содержит последнее снятое изображение. Левое-нижнее окно – бинарное изображение после применения операции Thresholding. Правое нижнее окно содержит шаблонное изображение маркера. Наконец, правое-верхнее окно содержит финальный график распределения метрики последней операции распознавания образов.

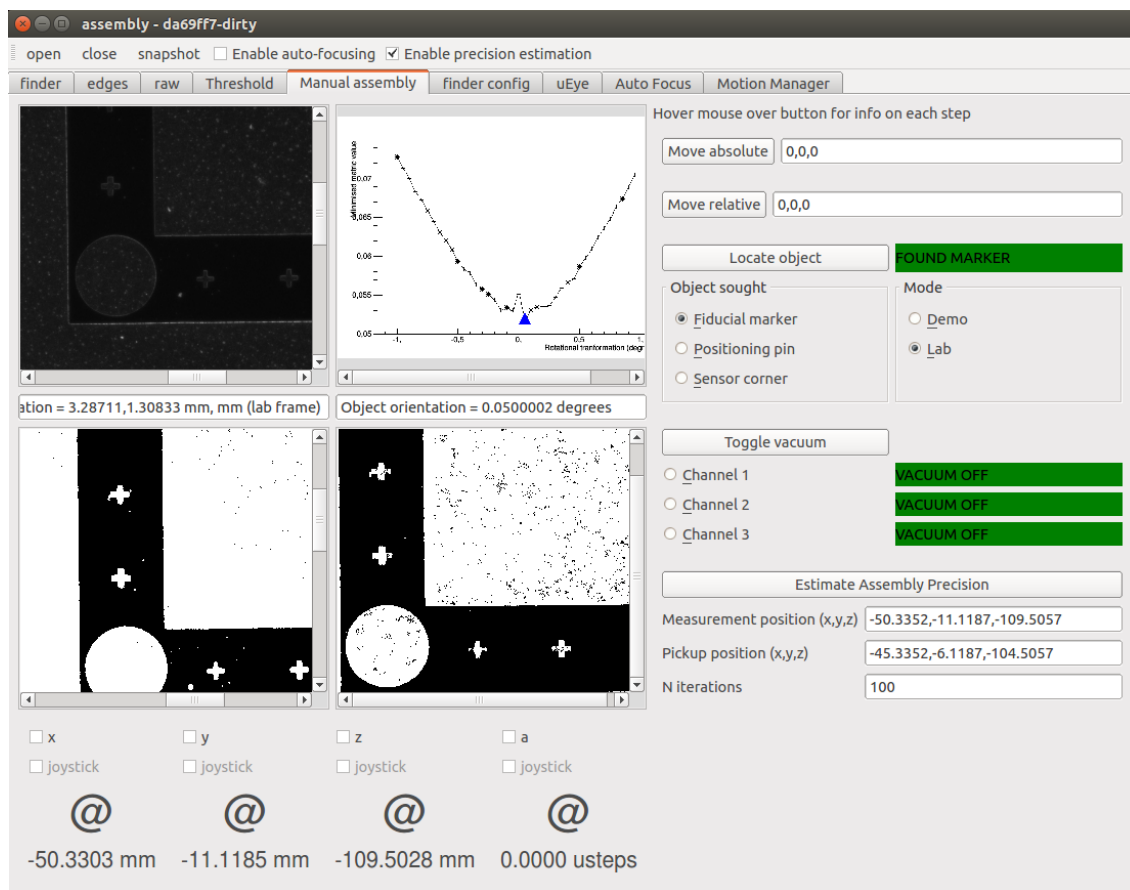


Рисунок 2.9 – Вид вкладки Assembly приложения PSAuto

В правой части вкладки расположен набор средств управления (перечисляя сверху вниз):

- Две кнопки для перемещения устройства захвата в абсолютные или относительные координаты, которые записываются в поля справа от них, соответственно.
- Набор «радиокнопок» для выбора типа шаблона маркера и режим работы распознавания образов.
- Набор «радиокнопок» для контроля релейной картой линий отрицательного давления. Справа от каждой кнопки отображается текущий статус соответствующей линии.
- Последний инструмент позволяет проводить набор тестов системы. Он состоит из нескольких форм для исходной информации для теста и, ожидаемо, кнопку начала теста. Тесты выполняются в полностью автоматическом режиме и могут длиться неограниченно долго.

Auto focus (от англ. auto focus – автофокус). Ещё один полезный инструмент приложения – способность проводить автоматическую фокусировку камеры. Данная функция играет очень важную роль в работе процедуры распознавания образов, так как от качества её проведения непосредственно зависит качество снимаемых изображений. Данная вкладка позволяет осуществлять две основные функции: найти положение камеры с лучшим фокусом и переместить камеру в это положение. Чтобы найти сфокусированное положение камеры необходимо пройти ряд шагов вдоль оси Z двигательной установки. Чем меньше шаг, тем точнее будет определено сфокусированное положение камеры. Идея работы данного алгоритма схожа с определением ориентации маркеров в алгоритме распознавания образов. Для каждого шага высчитывается и сохраняется определённое значение метрики, характеризующее степень «размытости» изображения, для чего используется функция библиотеки OpenCV `getImageBlur()`. Таким образом, сравнивая значения метрик каждого шага, программное обеспечение способно найти положение камеры, при котором получаемое изображение имеет наименьшее размытие.

Данный алгоритм также может использоваться и с другой целью. В силу того, что камера постоянно зафиксирована на роборуке, существует возможность измерения относительного расстояния вдоль оси Z до объекта под камерой. Данное применение возможно, например, для измерения толщины слоя клеящего вещества, собранного СРСП или контроля любых других метрических параметров модуля.

Motion Manager (от англ. motion manager – менеджер передвижений). Данная вкладка предоставляет контроль над всеми механическими движениями системы. Её функционал включает в себя:

- Независимый контроль вращения и движения по всем трём осям.
- Автоматическая калибровка.
- Слежением за статусом всех элементов двигательной подсистемы в режиме реального времени, включающие в себя четыре шаговых двигателей (один на вращение и три на движение по вдоль трёх осей).

Глава 3

Тестирование системы и сборка первого прототипа

Система автоматизированной сборки обладает рядом свойств с точки зрения точности сборки:

1. Повторяемость передвижений двигательной установки.
2. Повторяемость съёмки изображений.
3. Точность процедуры распознавания образов.
4. Возможные движения сенсора во время его захвата и отпускания с помощью системы отрицательного давления.

С целью изучения данных свойств был проведён ряд тестов системы.

Настоящие сенсоры будут иметь очень малую толщину (200 мкм), что делает их очень хрупкими. Даже несмотря на то, что для тестов могут быть использованы фиктивные сенсоры, толщина которых составляет 300 мкм, они всё ещё слишком хрупки для первых тестов, во время которых вероятность нештатных ситуаций заметно выше. По этой причине для первых тестов захвата использовались образцы из стекла. Они имеют такие же размеры и достаточно близки по механическим свойствам с кремниевыми сенсорами. Более того, они заметно дешевле и, следовательно, в случае неудачного теста потеря стеклянного образца не так критична, как фиктивного кремниевого сенсора. Несмотря на всё, сказанное выше, ни один из стеклянных образцов не был разрушен в ходе проведения тестовых работ.

Хотя тесты захвата кремниевых сенсоров не проводились в виду рисков, возможность получения некоторой информации об их поведение в данном тесте можно спрогнозировать без непосредственного проведения самого теста. Проведя полный набор тестов со стеклянными образцами и проанализировав их результаты, можно будет сделать выводы о том, как процесс захвата и выпуска влияет на точность/позиционирование образца. Позднее, после точной калибровки параллельности нижней поверхности устройства захвата относительно поверхности сборочной платформы и проведения необходимых тестов программного обеспечения, можно будет уточнить предсказанные результаты для кремниевых сенсоров с гораздо меньшими рисками неудачного исхода.

3.1 Тест точности процедуры распознавания образов

С целью исследования повторяемости и точности процедуры распознавания образов был проведён ряд следующих тестов. В ходе этих тестов исследуемые образцы (и стеклянные, и кремниевые) оставались неподвижными, поэтому ошибки, вносимые нежелательным ожидаемым смещением образца во время процедуры захвата, исключаются.

3.1.1 Распознавание образов на угле стеклянного образца

В самом первом тесте точности распознавания образов в качестве маркеров стеклянных образцов просто использовались их углы. Для теста использовались тонкие образцы стекла с углами, покрашенными серебряной краской для лучшего контраста на бинарном изображении (Рисунок 3.1).

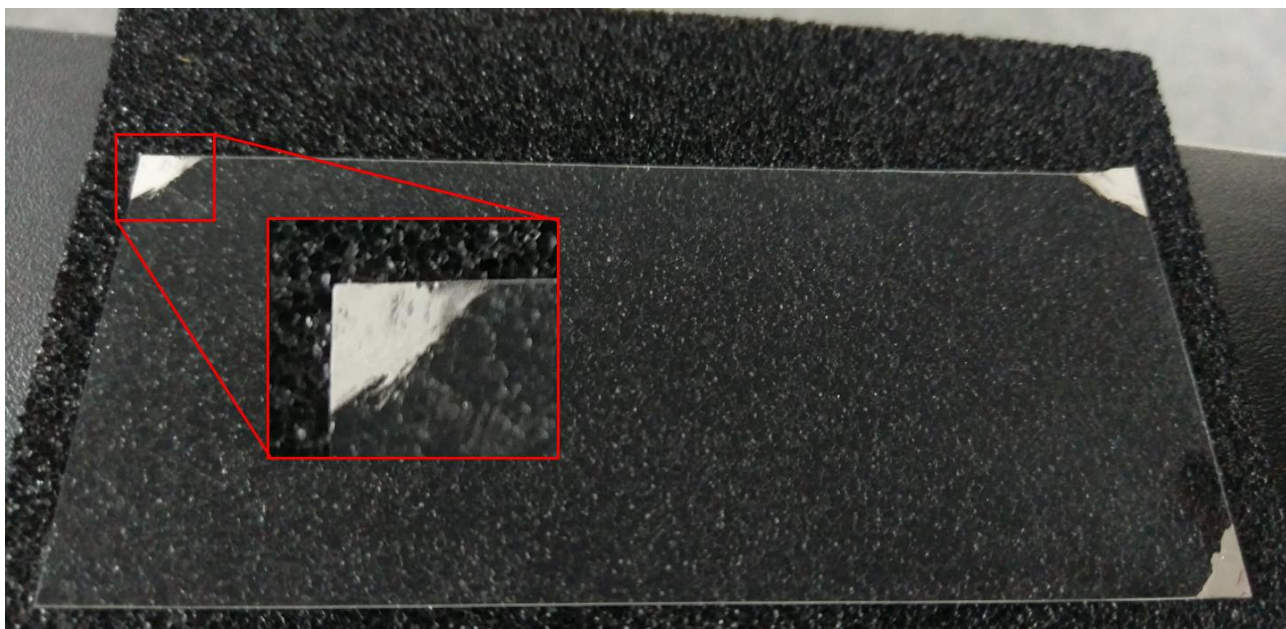


Рисунок 3.1 – Образцы стекла с углами, окрашенными серебряной краской

Пошаговое описание теста приведено ниже:

1. Передвинуться в позицию для съёмки изображения маркера.
2. Сделать снимок и провести процедуру распознавания образов.
3. Передвинуться на 5 мм по всем осям
4. Вернуться в позицию для съёмки изображения.
5. Сделать снимок и провести процедуру распознавания образов.
6. Сохранить данные данной итерации и перейти к следующей (к шагу 1).

После каждой итерации приложение сохраняет разницу между измеренными координатами маркера до и после перемещения роборуки. Итоговое распределение данных величин изображено на Рисунках 3.2, 3.3 и 3.4 для оси X, Y и α , соответственно. Тест состоял из 100 итераций, выполненных последовательно.

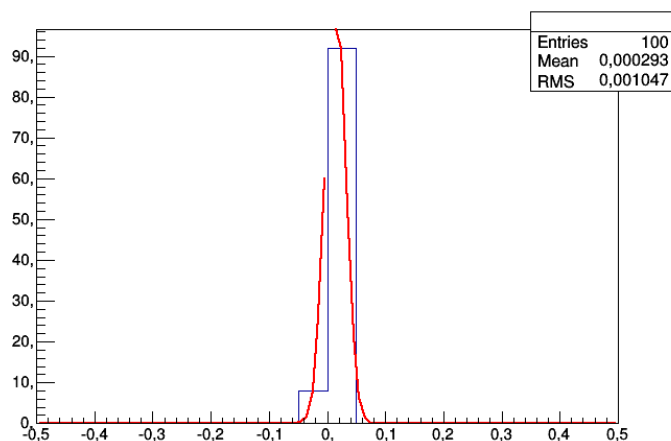


Рисунок 3.2 – Распределение разности по координате X до и после перемещения роборуки ($\Delta X \approx 1$ мкм)

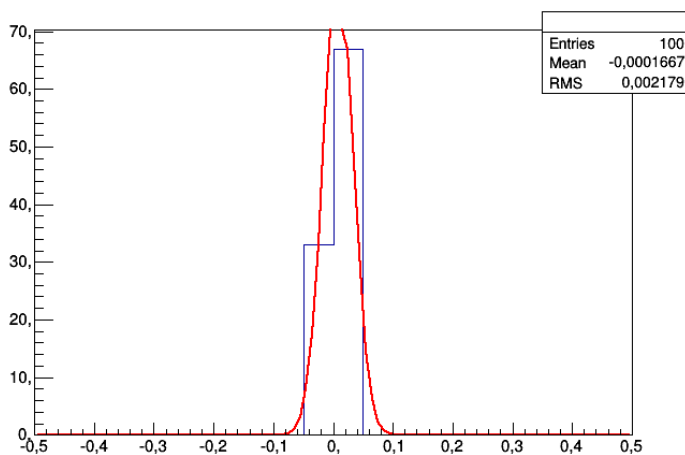


Рисунок 3.3 – Распределение разности по координате Y до и после перемещения роборуки ($\Delta Y \approx 2$ мкм)

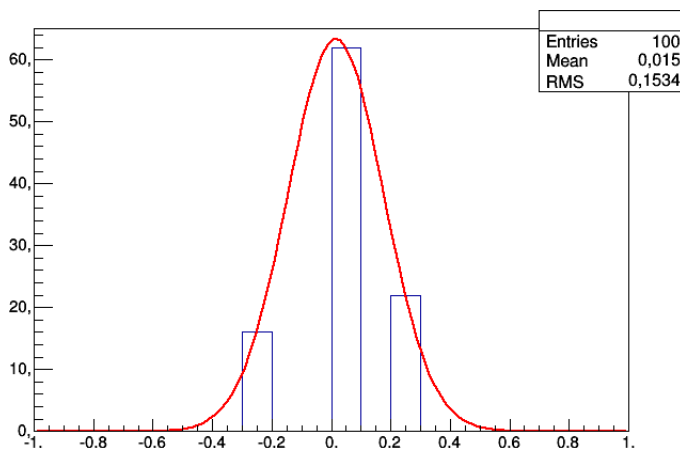


Рисунок 3.4 – Распределение разности значения определённого угла поворота между исследуемым изображением относительно шаблонного изображения до и после перемещения роборуки ($\Delta \alpha \approx 0,15$ градусов)

На Рисунках 3.2 и 3.3 можно видеть, что повторяемость измерения координат X , Y находится в пределах 1-2 мкм, в то время как измерения угла α показывает гораздо меньшую точность. На то есть несколько причин. Основная из них показана на Рисунке 3.5.



Рисунок 3.5 – Бинарное изображение окрашенного угла стеклянного образца после применения процедуры *Thresholding*

Поверхность серебряной краски не является достаточно ровной в масштабах микрометров. Из-за этих неровностей свет неравномерно отражается от поверхности, что приводит к тому, что на изображении угла в оттенках серого окрашенная поверхность в некоторых точках темнее поверхности (фона) под образцом. Результат данного эффекта можно наблюдать на Рисунке 3.5. Изображения такого рода имеют случайное распределение тёмных областей. Именно поэтому процесс распознавания образов выдаёт такие неточности, сравнивая исследуемое изображение со *случайным распределением* тёмных областей и шаблонное изображение с *постоянным распределением* тёмных областей. Более того, для набора достаточного числа статистических данных тесты такого рода содержат большое количество итераций (100 в данном случае), что соответствует нескольким часам работы установки. Этого времени достаточно для того, чтобы естественное освещение в лаборатории изменилось. Несмотря на принятые меры изоляции установки от естественного освещения, оптимальное пороговое значение операции *Thresholding* изменилось на 20 единиц (глубина цвета изображения в оттенках серого – 256 единиц) за время эксперимента.

3.1.2 Распознавание образов на маркере фиктивного сенсора

Аналогичный предыдущему тест был проведён с фиктивным сенсором, имеющем действительный маркер. Для удобства проведения теста перед его началом сенсор был ориентирован как можно ближе к положению, соответствующему $\alpha = 0$. На Рисунке 3.6 показан маркер фиктивного сенсора после применения операции Thresholding. Как видно из Рисунка 3.6, граница маркера практически идеальна (\pm один пиксель). Данный факт подтверждает обоснованность применения операции Thresholding ещё до проведения тестов.

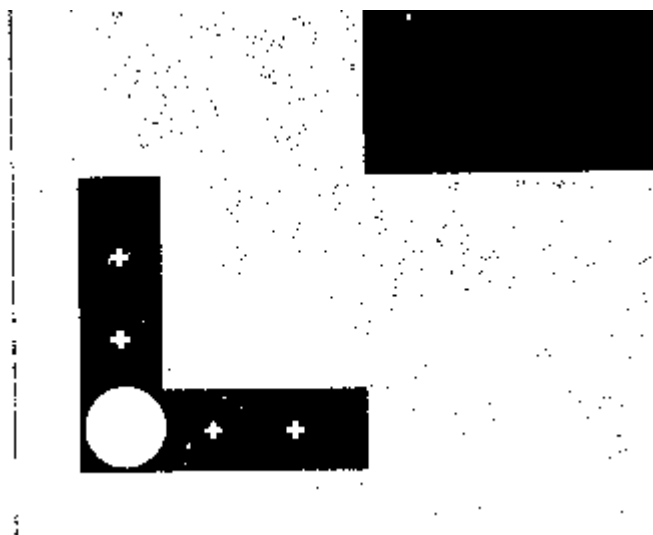


Рисунок 3.6 – Маркер фиктивного сенсора после проведения операции Thresholding

Распределение величин X , Y и α показывает намного лучшие результаты, чем у окрашенного серебряной краской угла, что вполне ожидаемо. Для координат X и Y точность составляет меньше микрона, что уже граничит с пределом разрешающей способности камеры. $\Delta\alpha$ в свою очередь стала на один порядок точнее, чем у окрашенного угла стеклянного образца – $\approx 0,02$ градуса.

Вид работы приложения во время теста изображён на Рисунке 3.7. На графике распознавания образов в точке, соответствующей 0 градусам, можно заметить неожиданный короткий пик, направленный вверх. Данный пик не является флуктуацией и не состоит только из одной точки на графике. С увеличением масштаба графика всё больше и больше точек начинают принимать участие в его формировании. Он не наблюдался в предыдущих тестах, так как шаг α был на порядок выше.

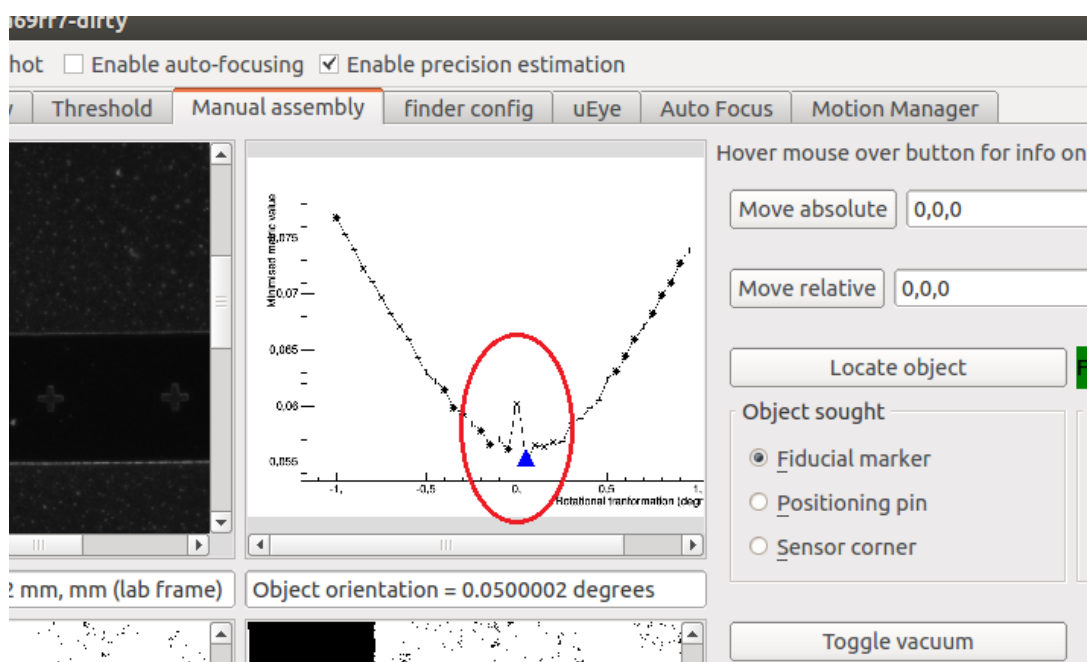


Рисунок 3.7 – Вид работы приложения во время проведение теста точности и повторяемости с фиктивным кремниевым сенсором

3.2 Тест захвата образца

Следующий ряд тестов был ориентирован на исследование нежелательного эффекта механического смещения образца во время его захвата и выпуска. В виду высокого риска механического повреждения сенсора использовались стеклянные образцы.

3.2.1 Тест захвата образца без помощи сборочной платформы

Самый первый тест захвата/выпуска был выполнен без помощи сборочной платформы. Его пошаговое описание приведено ниже:

1. Передвинуться в позицию съёмки изображения маркера.
2. Сделать снимок и провести процедуру.
3. Передвинуться в предварительную позицию захвата.
4. Передвинуться в позицию захвата.
5. Подать отрицательное давление на устройство захвата.
6. Передвинуться вверх на 5 см.
7. Передвинуться обратно вниз на 5 см.
8. Отключить подачу отрицательного давления на устройство захвата.
9. Передвинуться в предварительную позицию захвата.
10. Передвинуться в позицию съёмки изображения маркера.
11. Сделать снимок и провести процедуру

12. Сохранить данные текущей итерации и перейти к следующей (к шагу 1).

Перемещение сперва в предварительную позицию захвата перед непосредственным переходом к позиции захвата очень важно. Двигательная установка обеспечивает равную скорость движения по всем трём осям, поэтому, когда она получает команду для перемещения в какое-либо положение, то начинает двигаться одновременно с равной скоростью по всем трём осям к месту назначения. Как только необходимая координата на одной оси достигнута, двигательная установка, очевидно, перестает двигаться по этой оси, а движение по другим осям продолжается. Поэтому может возникнуть нежелательная ситуация, когда роботизированная рука достигает образца по оси Z , в то время как оси X и Y все равно будут двигаться, что может привести к повреждению или даже уничтожению образца. Чтобы предотвратить такую ситуацию, был добавлен шаг перехода в предварительную позицию захвата, которая отличается от позиции захвата только координатой Z – она на 5 см выше. Таким образом, из позиции предварительного захвата роборука перемещается строго вниз и безопасно касается образца для его захвата.

Результаты теста показали смещение образца, которое можно было заметить даже невооруженным глазом. Чтобы свести к минимуму это движение, было решено провести "тест касанием" – те же этапы, что и у теста захвата, но без подачи отрицательного давления. Такой тип теста может показать вклад касания к перемещению образца в тесте захвата. Результаты этого теста были очень похожи на результаты предыдущего теста, что означает, что непосредственно касание сенсора вносит наибольший вклад в смещение образца во время процедуры захвата. Интересным фактом этого движения является его характер. На Рисунке 3.8 можно увидеть зависимость движения образца по осям X и Y относительно номера итерации в тесте. Эти графики показывают, что движение не является случайным, а является более-менее постоянным как по значению, так и по направлению.

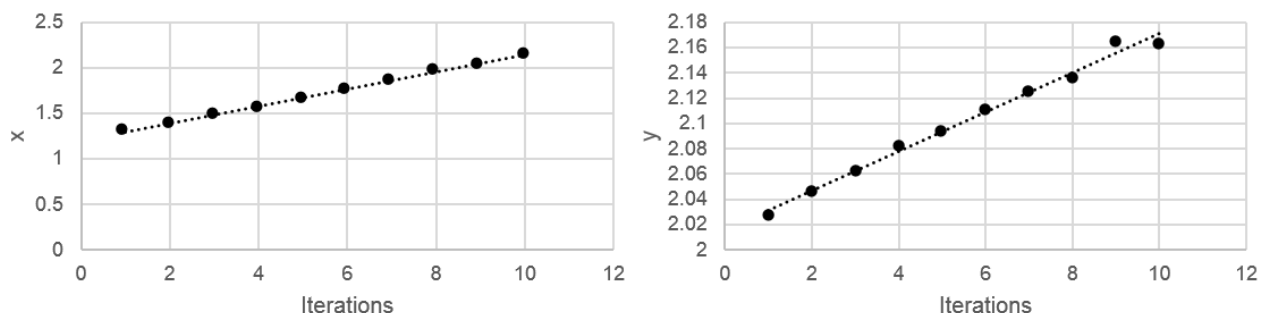


Рисунок 3.8 – Движения образца по осям X и Y относительно номера итерации в тесте касанием

Наиболее вероятной причиной такого поведения сенсора является антипараллельность поверхности устройства захвата и поверхности стола, на котором располагались образцы в данном тесте. Этот факт хорошо соответствует

постоянному смещению образца в одном направлении. К сожалению, очень сложно выровнять эти поверхности достаточно параллельно, чтобы движение образца было незначительным, так как в любом случае положение образца будет представлять из себя неустойчивое равновесие. Другой, более простой и эффективный, способ избежать этого движения – использовать сборочную платформу, которая может удерживать образец отрицательным, не допуская какого-либо их движения.

3.2.2 Тест захвата образца с помощью сборочной платформой

Пошаговое описание тест захвата образца с помощью сборочной платформой очень похож на тест без неё. Различием между ними является постоянная фиксация образца отрицательным давлением снизу и его отпускание только после включения подачи отрицательного давления устройством захвата. Другим словами происходит процесс "передачи" образца сборочной платформой устройству захвата.

1. Передвинуться в позицию съёмки изображения маркера.
2. Сделать снимок и провести процедуру.
3. Передвинуться в предварительную позицию захвата.
4. Передвинуться в позицию захвата.
5. Подать отрицательное давление на устройство захвата.
6. Отключить подачу отрицательного давления на сборочную платформу.
7. Передвинуться вверх на 5 см.
8. Передвинуться обратно вниз на 5 см.
9. Подать отрицательное давление на сборочную платформу.
10. Отключить подачу отрицательного давления на устройство захвата.
11. Передвинуться в предварительную позицию захвата.
12. Передвинуться в позицию съёмки изображения маркера.
13. Сделать снимок и провести процедуру
14. Сохранить данные текущей итерации и перейти к следующей (к шагу 1).

Распределение разности между измеренной координатой X маркера до и после перемещения роборуки показано на Рисунке 3.9.

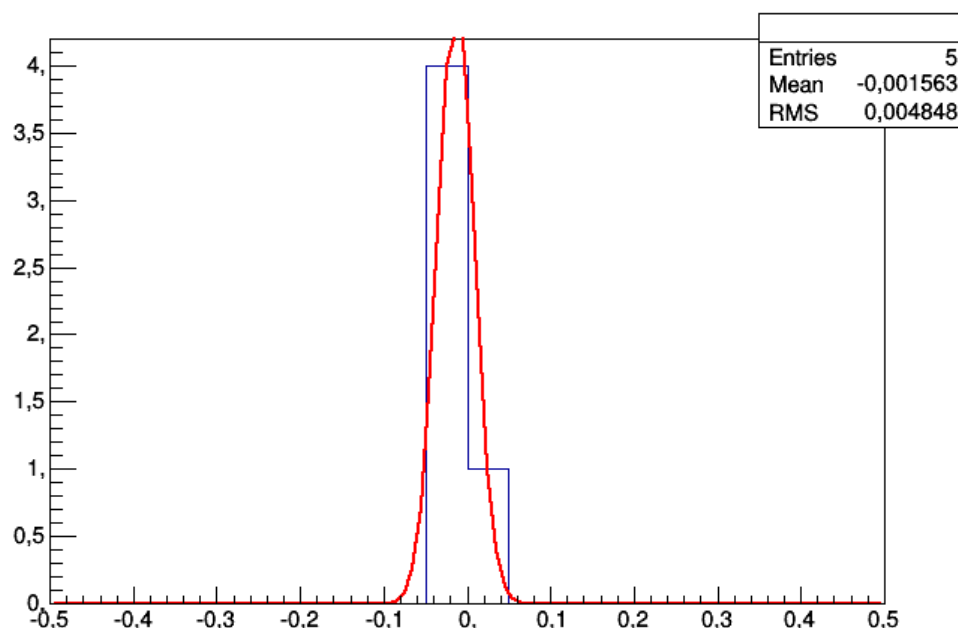


Рисунок 3.9 – Распределение разности между измеренной координатой X маркера до и после перемещения роборуки ($\Delta X \approx 4$ мкм)

Как можно видеть из Рисунка 3.9, $\Delta X \approx 4$ мкм, что очень близко к пределу процедуры распознавания образов (≈ 1 мкм) без касания образцы. Более того, Абсолютные значения измеренных координат X и Y показывают отсутствие какой-либо зависимости от номера итерации, как в тесте без сборочной платформы. Принимая во внимания полученные результаты, можно утверждать, что сборочная платформа способна фиксировать образцы достаточно плотно, чтобы пренебречь их нежелательными механическими смещениями.

3.3 Сборка первого прототипа

После всех выше упомянутых тестов было решено провести сборку первого прототипа модуля. Для проверки точности его сборки было принято решение использовать упрощенный алгоритм сборки, который обеспечивал бы максимальную точность только на одном из четырёх углов. Схематическое изображение первого прототипа модуля показано на Рисунке 3.10.

Пошаговое описание упрощённого процесса сборки описано ниже:

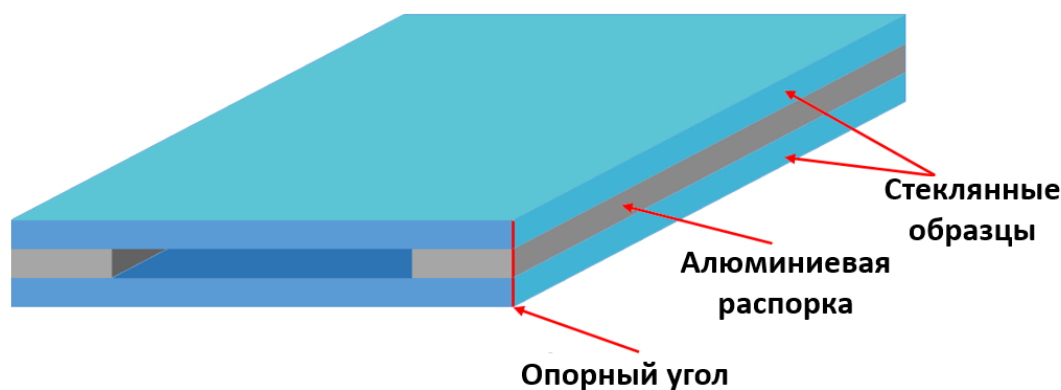


Рисунок 3.10 – Схематическое изображение первого прототипа модуля

1. *Подготовить верхний стеклянный образец.* Поместить верхний образец на сборочную платформу и зафиксировать его с помощью отрицательного давления. С помощью процедуры распознавания образов определить его ориентацию в пространстве. Так как необходимая точность определения планарной ориентации не обеспечивается путём его определения по одному маркеру, то для уточнения этого значения использовался следующий метод. Переместить камеру на 5 см вдоль края образца и измерить смещение края перпендикулярно движению. Пройденное расстояние и смещение образуют два катета прямоугольного треугольника, зная которые можно вычислить недостающий угол поворота между кромкой образца и осью X двигательной установки. Далее следует повернуть платформу на этот угол. Другими словами, сперва – грубая оценка угла ориентации образца с помощью процедуры распознавания образов, после – точное вычисления угла по теореме Пифагора. После того, как образец правильно сориентирован, необходимо сохранить координаты опорного угла образца и провести процедуру его захвата и подъёма.
2. *Подготовить алюминиевые распорки.* Повернуть сборочную платформу на 90 градусов. Поместить алюминиевые распорки в выемки сборочной платформы, зафиксировать их отрицательным давлением и сориентировать их параллельно оси X двигательной установки. Данная процедура делается аналогично верхнему образцу. Однако стоит упомянуть, что достижение высокой точности в ориентации распорок не настолько важно, как для сенсоров в модуле. Далее следует сохранить координаты опорного угла распорок. Сравнивая эти координаты с координатами опорного угла верхнего образца, переместить роботу в плоскости ХоУ для их совмещения.

3. *Произвести склеивание алюминиевых распорок с верхним образцом.* Нанести клеящее вещество на распорки и переместить вниз роборуку с прикреплённым к ней верхним образцом. Очень важно переместиться вниз на корректное расстояние. Данная тема была подробнее обсуждена в Главе 2. Далее необходимо выждать время высыхания быстросохнущего клеящего вещества (около 15 минут) и поднять полученную структуру (две алюминиевые распорки + верхний образец) со сборочной платформы, предварительно выключив подачу отрицательного давления в платформе.
4. *Подготовить нижний стеклянный образец.* Повернуть сборочную платформу на 90 градусов в противоположном направлении. Поместить нижний образец на сборочную платформу и зафиксировать его отрицательным давлением. По примеру верхнего образца провести ориентирование нижнего образца параллельно оси X двигательной установки. Измерить координаты опорного угла нижнего образца и сопоставить их с координатами верхнего. Переместить роборуку так, чтобы они совмещались.
5. *Финальное склеивание.* Нанести клеящее вещество на нижний образец и переместить роборуку вниз на корректную дистанцию, учитывая толщины распорок, стеклянных образцов и желаемого слоя клеящего вещества. После высыхания вещества процесс автоматизированной сборки можно считать оконченным.

Собранный прототип изображён на Рисунке 3.11.

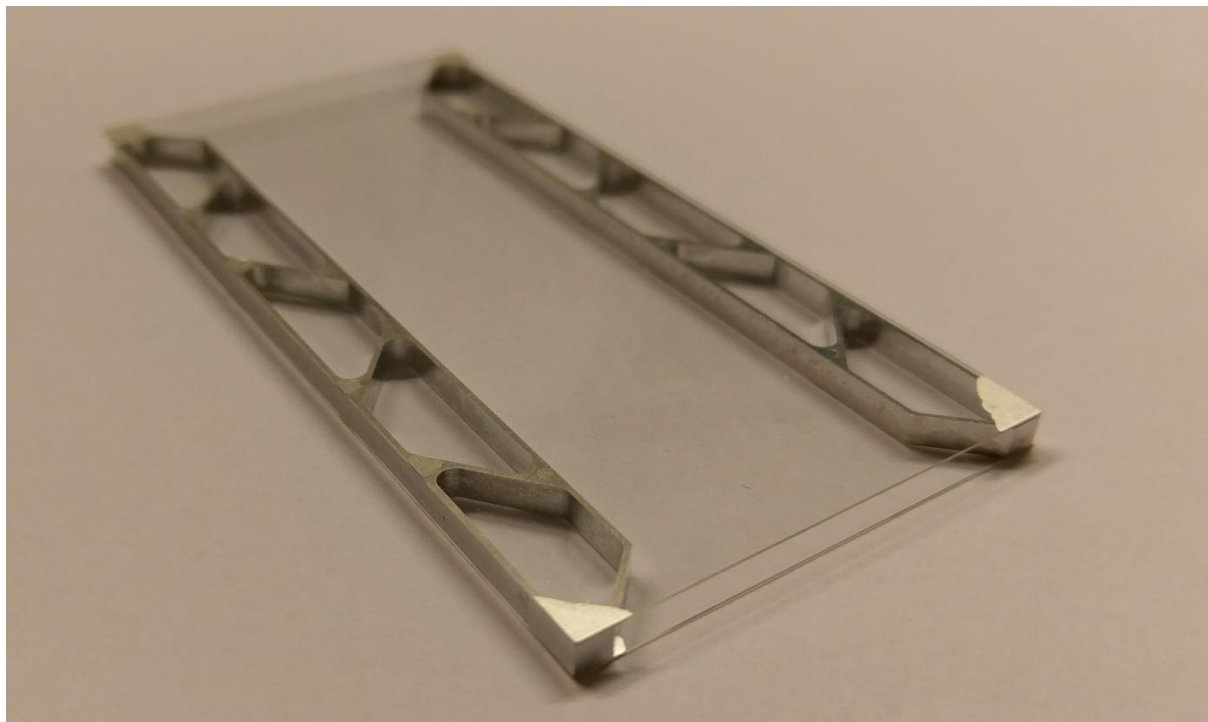


Рисунок 3.11 – Фото первого собранного прототипа PS модуля

Использованный упрощённый алгоритм автоматизированной сборки рассчитан на точное позиционирование только одного угла прототипа. Качество собранного прототипа выглядит многообещающим. Оно удовлетворяет всем требованиям, в том числе рассогласованность компонентов собранного прототипа укладывается в 20 мкм. На Рисунке 3.12 изображено фото опорного угла под микроскопом. Параметры угла с других направлений выглядят схоже, имея приблизительно одинаковую точность.

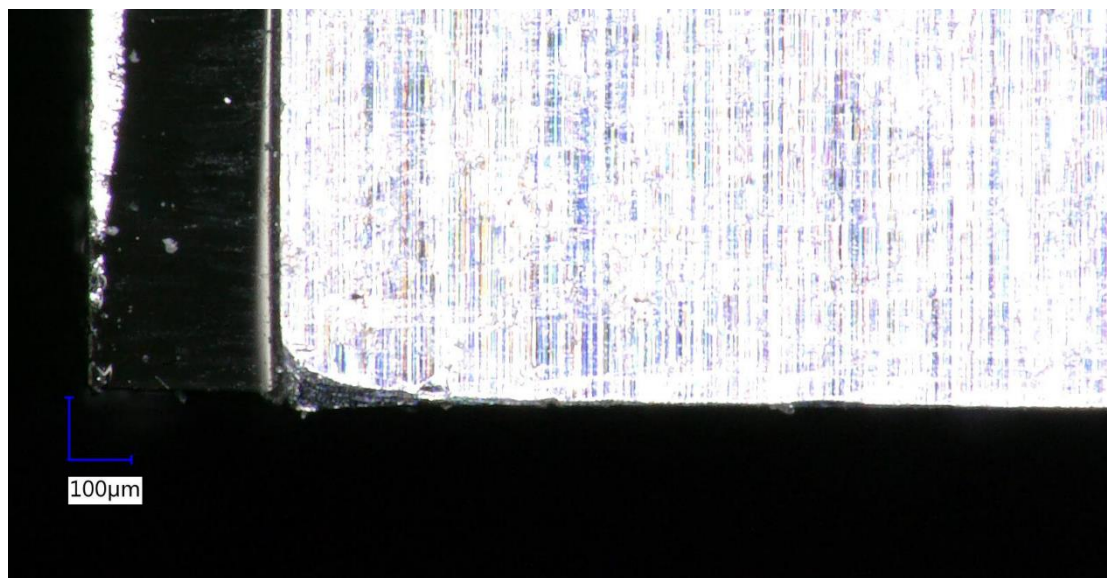


Рисунок 3.12 – Фото опорного угла собранного прототипа под микроскопом. Несогласованность компонентов находится в пределах 20 мкм

Собранный прототип подтвердил осуществимость системы автоматизированной сборки и показал её возможности, а также и моменты, на которые стоит обратить внимание в дальнейшем. Несмотря на то, что собранный прототип в качестве маркеров использовал окрашенные углы, а не точную литографию, как настоящие и фиктивные сенсоры, он показал очень хорошие результаты в плане точности сборки.

Заключение

В ходе моей работы над проектом автоматизированной сборки модулей CMS детектора второй стадии модернизации Большого Адронного Коллайдера были достигнуты следующие результаты:

- Внесены существенные модификации в приложение PSAuto, в том числе:
 - Интегрирован модуль управления преобразованием изображений, получаемых от камеры, что позволяет улучшить качество работы алгоритма распознавания образов.
 - Интегрирован гибкий и удобный метод управления системой отрицательного давления, согласующийся с шаблоном проектирования MVC.
 - Введены некоторые методы защиты приложения и системы от ошибок, в том числе и пользователя.
 - Интегрирован и протестирован модуль приложения, ответственный за автоматизированную сборку первого прототипа полностью автоматизированной сборки.
- Доказана возможность автоматизированной сборки в рамках указанных требований.
- Собран первый прототип модуля, показывающий потенциал системы.
- Была разработана, изготовлена, внедрена в систему и успешно протестирована сборочная платформа.
- Была исследована и успешно применена технология применения быстросохнущего вещества в процессе сборки.
- Было проведено множество тестов всей системы демонстрирующих возможности системы и предоставляемую ею точность сборки.

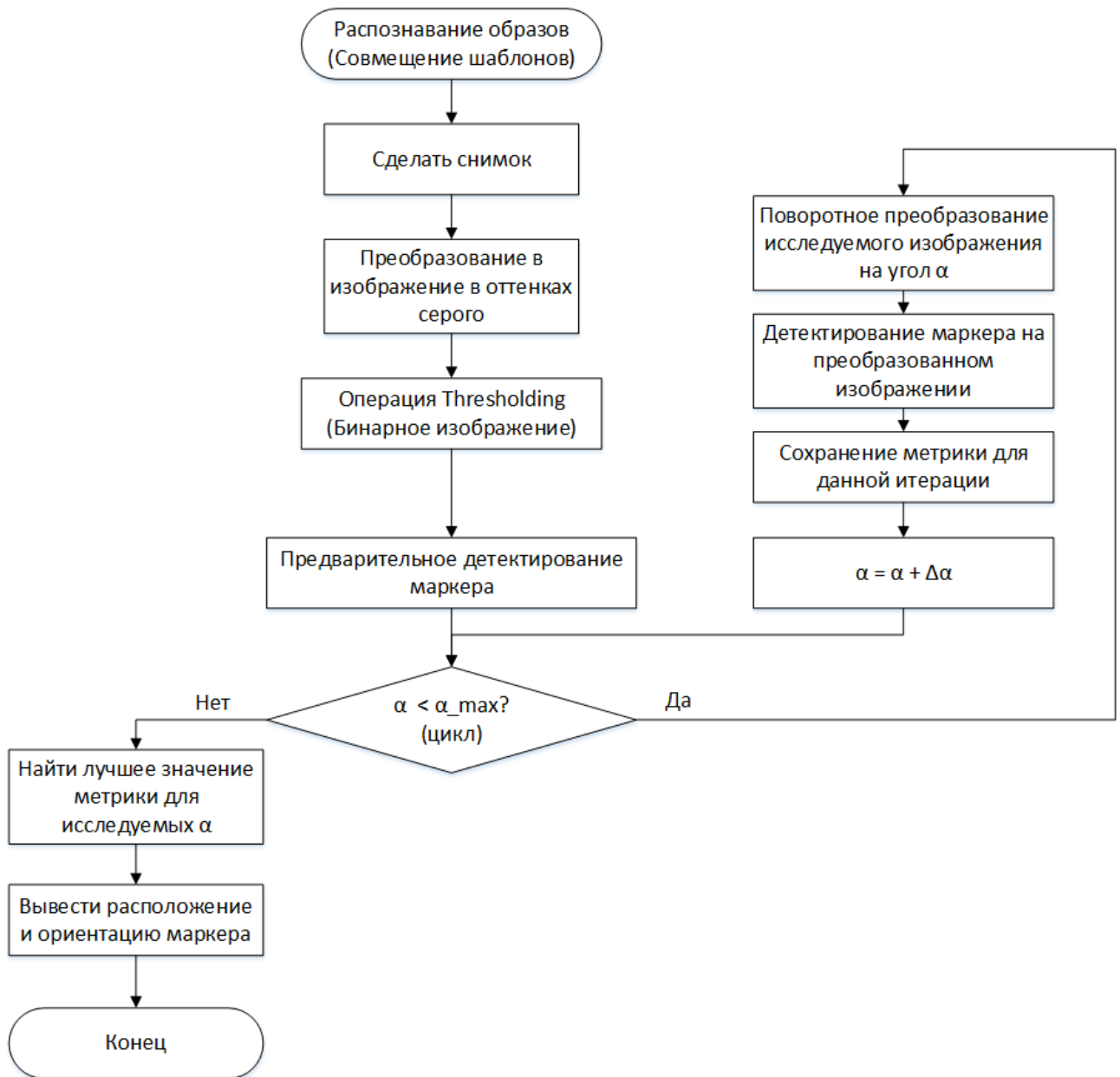
Ссылка на репозиторий с кодом приложения:

<https://github.com/bryksa/cmstkmodlab>.

Список использованных источников

1. "The Large Hadron Collider", CERN. [Electronic resource] – 2017. Mode of access: <https://home.cern/topics/large-hadron-collider> – Date of access: 15.06.2017.
2. "Technical Proposal For The Phase-II Upgrade Of The Compact Muon Solenoid" 2015. [Electronic resource]. Mode of access: <https://cds.cern.ch/record/2020886/files/LHCC-P-008.pdf> – Date of access: 15.06.2017.
3. Keaveney, J. M. "Commissioning of an automated assembly system for the PS modules of the CMS phase II upgrade outer tracker" / J. M. Keaveney, Hamburg, 2017. Mode of access: <https://github.com/keaveney/assemblyDoc/blob/master/bachelorprojectoutline.pdf> – Date of access: 15.06.2017.
4. Keaveny, J. M. "Towards an automated PS module assembly system" / J. M. Keaveny, A. Mussgiller, C. Muhl and A. Zuber, Hamburg, 2016.
5. "Data Sheet. Polytec EP 660.," 2017. [Electronic resource]. Mode of access: http://www.polytec.com/fileadmin/user_uploads/Polytec_EP_660_engl.pdf – Date of access: 15.06.2017.
6. "Technical Data Sheet. LOXEAL 31-42," 2017. [Electronic resource]. Mode of access: <http://www.loxeal.com/files/3142e.pdf> – Date of access: 15.06.2017.
7. "Wekem product catalog," 2007. [Electronic resource]. Mode of access: <http://www.jsj.lv/upload/content/wekem-english.pdf> – Date of access: 15.06.2017.
8. "Model-View-Controller," 2012. [Electronic resource]. Mode of access: <https://developer.apple.com/library/content/documentation/General/Conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html> – Date of access: 15.06.2017.
9. "About OpenCV," 2017. [Electronic resource]. Mode of access: <http://opencv.org/about.html> – Date of access: 15.06.2017.
10. Kaehler, A. Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library / A. Kaehler, G. Bradski Sebastopol: O'Reilly Media, 2016 – Date of access: 15.06.2017.
11. Pulli, K. "Realtime Computer Vision with OpenCV" / K. Pulli, A. Baksheev, K. Korniyakov and V. Eruhimov. Acm queue, c. 1-2, 2012
12. Agam, G. "Introduction to programming with OpenCV" / G. Agam. 2007. [Electronic resource]. Mode of access: <http://www.cs.iit.edu/~agam/cs512/lect-notes/opencv-intro/opencv-intro.html> – Date of access: 15.06.2017.

Приложение А



Приложение А – Блок схема процедуры распознавания образов.

Приложение Б

Исходный код метода распознавания образов приложения PSAuto:

```
void AssemblySensorMarkerFinder::findMarker_templateMatching(cv::Mat img, cv::Mat
img_clip_A)
{
    NQLog("AssemblySensorMarkerFinder") << "Finding Marker (Template Matching) here";

    NQLog("AssemblySensorMarkerFinder") << ", number of image channels =
"<<img_clip_A.channels() ;

    Point matchLoc_1, matchLoc_2, matchLoc_final;
    Mat result_1, result_2;
    Mat img_copy = img.clone();

    //Greyscale images
    Mat img_copy_gs(img_copy.size(), img_copy.type());
    Mat img_clip_A_gs(img_clip_A.size(), img_clip_A.type());

    if (img.channels()> 1){
        //convert color to GS
        cvtColor( img_copy,  img_copy_gs,  CV_BGR2GRAY );
    }else{
        img_copy_gs = img_copy.clone();
    }

    if (img_clip_A.channels()> 1){
        //convert color to GS
        cvtColor( img_clip_A, img_clip_A_gs, CV_BGR2GRAY );
    }else{
        img_clip_A_gs = img_clip_A.clone();
    }

    //Binary images
    Mat img_copy_bin(img_copy_gs.size(), img_copy_gs.type());
    Mat img_clip_A_bin(img_clip_A_gs.size(), img_clip_A_gs.type());

    //Apply thresholding
    cv::threshold(img_copy_gs, img_copy_bin, generalThreshold, 255,
cv::THRESH_BINARY);
    cv::threshold(img_clip_A_gs, img_clip_A_bin, 90, 255, cv::THRESH_BINARY);

    // img_copy_bin = img_copy_gs.clone();
    // img_clip_A_bin = img_clip_A_gs.clone();

    std::string filename_img_bin = Config::CMSTkModLabBasePath +
"/share/assembly/Sensor_bin.png";
```

```

std::string filename_clip_A_bin = Config::CMSTkModLabBasePath +
"/share/assembly/clip_A_bin.png";

cv::imwrite(filename_img_bin, img_copy_bin);
cv::imwrite(filename_clip_A_bin, img_clip_A_bin);

QString filename_img_bin_qs = QString::fromStdString(filename_img_bin);
QString filename_clip_A_bin_qs = QString::fromStdString(filename_clip_A_bin);

//emit updateImage(4, filename_img_bin_qs);
//emit updateImage(5, filename_clip_A_bin_qs);

NQLog("AssemblySensorMarkerFinder") << "Finding Marker (Template Matching),
updated images in view" ;

/// Localizing the best match with minMaxLoc
double FOM, FOM_inc = 1000.0, minVal, maxVal; Point minLoc; Point maxLoc;
double thetas[40], FOMs[40], best_theta;

//create result matrix to hold correlation values
int result_cols = img_copy_bin.cols - img_clip_A.cols + 1;
int result_rows = img_copy_bin.rows - img_clip_A.rows + 1;

// result_1.create( result_rows, result_cols, CV_32FC1 );
result_1.create( result_rows, result_cols, CV_32FC1 );

int match_method = 1;
matchTemplate( img_copy_bin, img_clip_A_bin, result_1, match_method);

NQLog("AssemblySensorMarkerFinder") << "Finding Marker (Template Matching), pre-
matching routine" ;

minMaxLoc( result_1, &minVal, &maxVal, &minLoc, &maxLoc, Mat() );

/// For SQDIFF and SQDIFF_NORMED, the best matches are lower values. For all the
other methods, the higher the better
if( match_method == CV_TM_SQDIFF || match_method ==
CV_TM_SQDIFF_NORMED ) {
    matchLoc_1 = minLoc;
    FOM = minVal;
} else {
    matchLoc_1 = maxLoc;
    FOM = maxVal;
}

NQLog("AssemblySensorMarkerFinder") << "Finding Marker (Template Matching), post-
matching routine" ;

//circle( img_gs, Point( matchLoc_1.x + (img_clip_gs.cols/2.0) , matchLoc_1.y +
(img_clip_gs.rows/2.0) ), 10, Scalar(255,0,0) );

```

```

//circle( img, Point( matchLoc_1.x , matchLoc_1.y ), 30, Scalar(255,0,0) );
//rectangle( img, matchLoc_1, Point( matchLoc_1.x + img_clip_A_bin.cols , matchLoc_1.y
+ img_clip_A_bin.rows ), Scalar(255,0,0), 2, 8, 0 );

NQLog("AssemblySensorMarkerFinder") << matchLoc_1.x << " " << matchLoc_1.y;

int i = 0;
int color = 200;

for (float theta = -1; theta < 1; theta = theta + 0.05){
    // for (float theta = -64.0; theta < 64.0; theta = theta + 3.2){
    // for (float theta = -180.0; theta < 180.0; theta = theta + 9.0){

    // Point2f src_center(img_gs_copy.cols/2.0F, img_gs_copy.rows/2.0F);
    Point2f src_center( matchLoc_1.x + (img_clip_A_bin.cols/2) , matchLoc_1.y +
(img_clip_A_bin.rows/2) );

    Mat rot_mat = getRotationMatrix2D(matchLoc_1, theta, 1.0);
    Mat dst;
    cv::Scalar avgPixelIntensity = cv::mean( img_copy_bin );
    warpAffine(img_copy_bin, dst, rot_mat, img_copy_bin.size(), cv::INTER_CUBIC,
cv::BORDER_CONSTANT, avgPixelIntensity);

    std::string filename_rotated_base = cacheDirectory2_ + "/Rotation_result_";
    std::ostream ss;
    ss << theta;
    std::string theta_str = ss.str();
    std::string filename_rotated = filename_rotated_base + theta_str + ".png";
    cv::imwrite(filename_rotated, dst);

    //create result matrix to hold correlation values
    result_cols = img_copy_bin.cols - img_clip_A_bin.cols + 1;
    result_rows = img_copy_bin.rows - img_clip_A_bin.rows + 1;

    result_2.create( result_rows, result_cols, CV_32FC1 );
    matchTemplate( dst, img_clip_A_bin, result_2, match_method );

    // normalize( result, result, 0, 1, NORM_MINMAX, -1, Mat() );
    minMaxLoc( result_2, &minVal, &maxVal, &minLoc, &maxLoc, Mat() );

    /// For SQDIFF and SQDIFF_NORMED, the best matches are lower values. For all the
other methods, the higher the better
    if( match_method == CV_TM_SQDIFF || match_method ==
CV_TM_SQDIFF_NORMED ) {
        matchLoc_2 = minLoc;
        FOM = minVal;
    } else {
        matchLoc_2 = maxLoc;
        FOM = maxVal;
    }
}

```

```

    }

    std::cout << std::setprecision(10);

    std::cout <<"theta = "<< theta << ", FOM = "<< FOM<< std::endl;

    thetas[i] = theta;
    FOMs[i] = FOM;
    i++;

    if (FOM < FOM_inc ) {

        FOM_inc = FOM;
        matchLoc_final = matchLoc_2;
        best_theta = theta;
    }

    color = 50 + 200*theta;
}

cv::Mat img_raw = img.clone();
rectangle( img, matchLoc_final, Point( matchLoc_final.x + img_clip_A_bin.cols ,
matchLoc_final.y + img_clip_A_bin.rows ), Scalar(color,color-50,color+50), 2, 8, 0 );

cv::Rect rectangle(matchLoc_final, Point( matchLoc_final.x + img_clip_A_bin.cols ,
matchLoc_final.y + img_clip_A_bin.rows ) );

TCanvas *c1 = new TCanvas("c1","Rotation extraction",200,10,700,500);

if (thetas[0] && FOMs[0]){
    TGraph *gr = new TGraph(40, thetas, FOMs);

    //gr->Fit("pol6");
    gr->Draw("AC*");
    gr->GetHistogram()->GetXaxis()->SetTitle("Rotational tranformation (degrees)");
    gr->GetHistogram()->GetYaxis()->SetTitle("Minimised metric value");
    gr->GetHistogram()->SetTitle("");

    TGraph *gr_bestmatch = new TGraph(1);
    gr_bestmatch->SetPoint(0, best_theta, FOM_inc);
    gr_bestmatch->SetMarkerColor(2);
    gr_bestmatch->SetMarkerStyle(22);
    gr_bestmatch->SetMarkerSize(3);

    gr_bestmatch->Draw("PSAME");

    std::string filename_canvas = cacheDirectory1_ + "/RotationExtraction.png";
    c1->SaveAs(filename_canvas.c_str());
    QString filename_canvas_qs = QString::fromStdString(filename_canvas);

    emit updateImage(2, filename_canvas_qs);
}

```

```

std::string filename = cacheDirectory1_ + "/PatRec_TM_result.png";
cv::imwrite(filename, img);

QString filename_qs = QString::fromStdString(filename);
QString filename_template = QString::fromStdString(filename_clip_A_bin);
QString filename_master = QString::fromStdString(filename_img_bin);

emit updateImage(1, filename_qs);

emit foundSensor(1);
emit getImageBlur(img_raw, rectangle);

emit updateImage(3, filename_template);
emit updateImage(4, filename_master);


//work out match location in field of view
// the origin of the FOV coordinate system is the top left corner
//the match loction (centre of the template) is calculated in mm
//this should be enough for postion correction with moverealtive()
//the dimensions of the image in the lab frame need
//to be measured precisely somehow. For now taking width = 5mm
//and height = 4mm

matchLoc_x_lab = (matchLoc_final.x + (img_clip_A_bin.cols/2) ) * (5.0/img.cols); // need
to add the current X pos of the lang
matchLoc_y_lab = (matchLoc_final.y + (img_clip_A_bin.rows/2) ) * (4.0/img.rows); //
need to add the current Y pos of the lang

//update line edits in view

emit reportObjectLocation(1, matchLoc_x_lab, matchLoc_y_lab, best_theta);

}

```