**ОГЛАВЛЕНИЕ**

[**ВВЕДЕНИЕ** 4](#_Toc470105231)

[**1.** **Предпроектное исследование** 5](#_Toc470105232)

[**1.1.** **Общая информация по навигационным системам** 5](#_Toc470105233)

[**1.2.** **Оптические навигационные системы** 7](#_Toc470105234)

[**2.** **Техническое задание** 11](#_Toc470105235)

[**2.1.** **Общие сведения** 11](#_Toc470105236)

[**2.2.** **Назначение и цели создания системы** 11](#_Toc470105237)

[**2.2.1.** **Назначение и область применения** 11](#_Toc470105238)

[**2.2.2.** **Цель и задачи** 11](#_Toc470105239)

[**2.3.** **Характеристика объектов** 12](#_Toc470105240)

[**2.4.** **Требования к системе** 12](#_Toc470105241)

[**2.4.1.** **Требования к надежности** 12](#_Toc470105242)

[**2.4.2.** **Требования к функциональным характеристикам** 12](#_Toc470105243)

[**2.4.3.** **Технологические требования** 13](#_Toc470105244)

[**2.4.4.** **Требования по техническому обслуживанию** 14](#_Toc470105245)

[**2.4.5.** **Требования к программной документации** 14](#_Toc470105246)

[**2.4.6.** **Требования к климатическим условиям эксплуатации** 14](#_Toc470105247)

[**3.** **Концептуальное проектирование** 15](#_Toc470105248)

[**3.1.** **Методика тестирования системы** 15](#_Toc470105249)

[**3.2.** **Концептуальная модель модуля визуализации** 17](#_Toc470105250)

[**3.3.** **Алгоритм разрабатываемой программы** 19](#_Toc470105251)

[**4.** **Рабочее проектирование системы** 20](#_Toc470105252)

[**4.1.** **Библиотека OpenCV** 20](#_Toc470105253)

[**4.2.** **Поиск сферических маркеров** 21](#_Toc470105254)

[**4.3.** **Получение карты неравенства и расчет координат** 26](#_Toc470105255)

[**4.4.** **Визуализация полученных данных** 31](#_Toc470105256)

[**ЗАКЛЮЧЕНИЕ** 33](#_Toc470105257)

[**СПИСОК ЛИТЕРАТУРЫ:** 34](#_Toc470105258)

[**ПРИЛОЖЕНИЕ 1** 35](#_Toc470105259)

# **ВВЕДЕНИЕ**

В этой работе описывает процесс построения пользовательского стереозрения для стереопроцессов реального времени, от выбора камер до перепроектирования 3-D облака точек. Целью данной работы является разработка и анализ автоматизированной хирургической навигационной системы с использованием машинного зрения. Эта расчетно-пояснительная записка содержит четыре главы, описывающие проделанную работу. В первой главе рассматриваются типы хирургических навигационных систем, их основные применения, делается анализ аналогов системы и сравнение их характеристик. Во второй главе формулируется техническое задание на разработку программно-аппаратного комплекса, приводятся основные сведения и требования к системе. В третьей части описываются концептуальные методы моделирования, которые использовались при разработке программно-аппаратного комплекса. Четвертая глава описывает основные инструменты работы для программирования, а также функции, которые были использованы при написании кода программы. В заключении представлен результат проделанной работы, а также список используемой литературы.

1. **Предпроектное исследование**

В настоящее время задача улучшения качества проведения хирургических операций является одной из важнейших в российской медицине. Внедрение современного оборудования и программного обеспечения позволяет сократить время проведения сложных многочасовых операций, а также ускорить время постоперационного восстановления пациента. Одной из областей медицины, которая направлена на то, чтобы минимизировать вмешательство в организм пациента и уменьшить степень травмирования тканей, является - малоинвазивная хирургия. В настоящее время данная область в значительной степени развивается за счет использования навигационных хирургических систем. К сожалению, системы такого типа производится за рубежом, и не каждое медицинское учреждение может позволить себе его приобретение, а также переобучение персонала для работы с ним.

Целью данной работы является разработка и исследование отечественного аналога хирургической навигационной системы. Реализация поставленной цели позволит уменьшить затраты на приобретение, внедрение и поддержку данного программно-аппаратного комплекса в медицинские учреждения. В то же время реализация отечественной хирургической навигационной системы позволит упростить процесс переобучения и использования за счет русификации программного обеспечения, а также русификации требуемой документации.

* 1. **Общая информация по навигационным системам**

Хирургическая навигационная система - это программно-аппаратный комплекс, позволяющая врачу отслеживать положение хирургического инструмента относительно оперируемого объекта в реальном времени. Выбор метода или сочетания интраоперационной навигации основывается на точности позиционирования и скорости получения актуальных данных.В настоящий момент, хирургические навигационные системы применяются в таких областях медицины, как: нейрохирургия, ЛОР и ЧЛХ – хирургия, спинальная и травматологическая хирургия, ортопедическая хирургия.

Использование данных систем позволяет:

1. Добиться лучших клинических результатов операции
2. Тщательно планировать операцию и контролировать ее ход, понижая количество ошибок
3. Снижает операционную травму, так как позволяет подойти к оперируемому объекту по наикротчайшему и безопасному пути
4. Уменьшить время реабилитации пациента, уменьшить затраты на фармакотерапию, улучшает качество жизни в послеоперационном периоде

Обобщенная принципиальная схема типичной хирургической системы представлена на рисунке 1. В нее входят следующие подсистемы: подсистема регистрации положения и подсистема визуализации.

**Хирургическая навигационная система**

**Подсистема регистрации положения**

**Подсистема визуализации**

**Данные предоперационного исследования**

**Интерфейс обмена данными**

Рис.1 Обобщённая структура хирургической навигационной системы

Хирургические навигационные системы классифицируются следующим образом:

1. Оптические навигационные системы
2. Электромагнитные навигационные системы
3. Акустические навигационные системы
4. Механические навигационные системы

Характеристики данных хирургических систем представлены в таблице 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Механические** | **Оптические** | **Электромагнитные** | **Акустические** |
| Точность | 0,1-2,5мм | 0,1-2,5мм | ≈1,0-2,0мм | ≈1мм |
| Разрешение |  | ≈0,01мм |  | ≈0,1мм |
| Возможные помехи | Физические помехи | Нагреты тела | Магнитные поля, металлические объекты | Сырость, |
| Пассивные /Активные | Пассивные | Пассивные и активные | Активные | Активные |
| Примеры | Faro Arm, NeuroNavigator | Struker, Kick | Polhemus, Aurora | Sonic Wand |

Таблица 1. Характеристики навигационных систем

* 1. **Оптические навигационные системы**

В настоящее время наибольшее распространение получили оптические навигационные системы. Принцип работы данных систем базируется на регистрации положения видимых объектов – маркеров, по которым определяется положение и ориентация дистального наконечника хирургического инструмента.

Развитие компьютерных систем позволило существенно увеличить скорость детектирования маркеров. К достоинствам таких систем можно отнести:

1. Меньшая стоимость системы
2. Доступность комплектующих
3. Может детектировать несколько видимых объектов, не увеличивая цены конструкции системы
4. Инструмент, более легкий, по сравнению с другими навигационными системами

Основная сложность, возникающая при разработке данных систем, является подсистема регистрации маркеров, так как на результат могут влиять такие факторы, как:

1. Разная освещенность
2. Перекрытие
3. Зашумленность изображений
4. Деформации

Поэтому изображения требуют предварительной обработки. Также чаще всего для ускорения трудоемких вычислений используют распараллеливание. Итогами вычислений является координаты маркеров в пространстве. Имея информацию о геометрии инструмента, а также взаимное расположение маркеров и инструмента, можно воспроизвести положение инструмента в пространстве.

Оптическая навигационная система состоит из стойки, на которой закреплен оптический блок, состоящий из двух камер и лазерного указателя; монитора, для визуализации положения хирургического инструмента и блока управления. Пример такой станции изображен на рисунке 2.



Рис. 2. Оптическая хирургическая навигационная система

В свою очередь, оптические навигационные системы подразделяются на активную и пассивную технологии навигации. В пассивных системах маркеры, находящиеся на инструменте, излучают инфракрасные волны (Рисунок 3а), а в активных - излучение генерирует система, волны отражаются от маркеров и приходят в детектирующую систему (Рисунок 3б). После этих действий, вычисляется положение маркеров в системе.

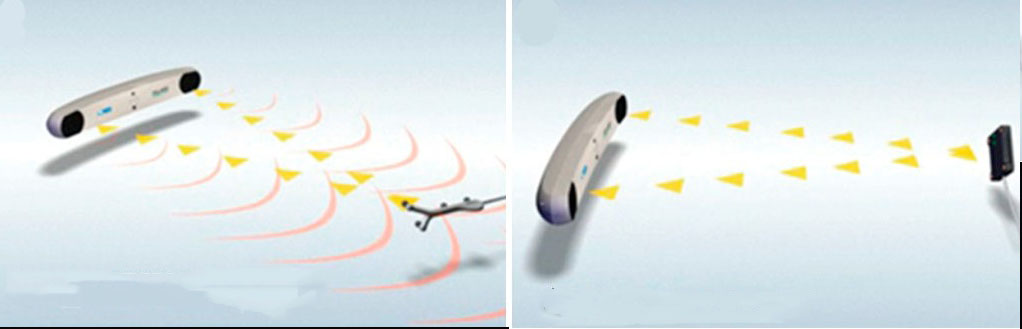


Рис. 3. а) Пассивная система

б) Активная система

Активная технология навигации имеет ряд преимуществ перед пассивной:

1. Оборудование и расходные материалы для активной навигации дешевле, чем для пассивной.
2. Сигнал, от используемого светодиода, лучше локализуется камерой, что дает преимущество в точности навигации более чем в 4 раза. (R. Khadem, 2000)
3. Меньшие габариты у инструмента, а форма более обтекаема.

Исходя из данных, описанных ранее, можно сделать вывод, что разрабатываемая хирургическая система должна иметь схожие характеристики с мировыми аналогами при меньшей стоимости оборудования и программного обеспечения. Она должна поддерживать загрузку индивидуальной модели пациента, чтобы учесть персональные особенности строения. Также должна быть предусмотрена возможность интегрирования с другими устройствами.

1. **Техническое задание**
   1. **Общие сведения**

Наименование системы: Автоматизированная хирургическая навигационная система

* 1. **Назначение и цели создания системы**
     1. **Назначение и область применения**

Данная автоматизированная хирургическая навигационная система – программно-аппаратный комплекс, предназначенный для:

* Проведения сложных хирургических операций в области вертебрологии
* Комплексной визуализации при проведении малоинвазивных хирургических операций

Система должна помочь хирургам в области вертебрологии и нейрохирургии, уменьшая время на проведение операции, а также пост операционное выздоровление пациента. Она позволит отслеживать расположение инструмента относительно индивидуальных анатомических особенностей пациента в реальном времени, тем самым облегчая вынесение решения в ходе оперирования. Областью применения данных систем является клинические центры, больницы и другие здравоохранительные центры.

* + 1. **Цель и задачи**

Основной целью проектирования данной системы является:

* Создание отечественной хирургической навигационной системы для использования в клинико-диагностических центрах, больницах и других учреждениях здравоохранения при проведении сложных хирургических и диагностических вмешательств, в том числе малоинвазивных.

В процессе выполнения работы должны быть решены следующие задачи:

* Разработка программного обеспечения для параллельной регистрации видео со стереосистемы
* Определение сферического маркера с точностью 1.0±0.5 мм
* Моделирование и визуализация положения инструмента и его перемещения
  1. **Характеристика объектов**

Объектом разработки является система, состоящая из:

* 2 камер для регистрации изображений
* Программного обеспечения
* Системы для управления и программирования
* Объектов детектирования
  1. **Требования к системе**
     1. **Требования к надежности**

Доступ к автоматизированной системе должен осуществляться в соответствии с должностью, занимаемой пользователем

Должна вестись история действий, совершенных над системой.

В данной автоматизированной системе должен быть предусмотрен резервный источник питания, подключающийся при авариной ситуации (сбой в электросети, выход из строя блока питания) во время оперирования пациента.

Хирургическая навигационная система должна обеспечивать точные измерения в зависимости от области ее применения, поддерживать регулировку данных в течении всего срока службы.

* + 1. **Требования к функциональным характеристикам**

Программное обеспечение, используемое на данной хирургической навигационной системе должно выполнять следующие функции:

* Функция калибровки стереосистемы
* Функция детектирования объектов, в том числе после их замены по истечению срока эксплуатации
* Функция загрузки индивидуальной модели анатомии пациента
* Функция просмотра положения хирургического инструмента относительно оперируемого объекта
* Функция сохранения требуемых хирургических данных, которые в дальнейшем могут быть использованы
* Поддержка справочной системы
* Возможность подключения стороннего оборудования или программных пакетов, при наличии необходимого доступа
  + 1. **Технологические требования**

Ниже представлен перечень требований, предъявляемый к навигационной системе.

Автоматизированная система хирургической навигации должна иметь две камеры с разрешением не ниже 640х480 точек при 30fps, оснащенных ПЗС-матрице с одинаковым размером пикселя, а также с одинаковым фокусным расстоянием. Для вычисления система должна обладать вычислительным блоком, имеющий необходимые характеристики для поддержки необходимого программного обеспечения, а также для выполнения всех вычислительных операций. Устройство должно обладать блоком питания с входным напряжением 220В и током 5А. Для вывода информации и управлением программой во время операции должен быть предусмотрен сенсорный дисплей.

* + 1. **Требования по техническому обслуживанию**

Техническая поддержка программного и аппаратного обеспечения может производится, как удаленно по телефону или электронной почте, так и с выездом обеспечивающего персонала.

* + 1. **Требования к программной документации**

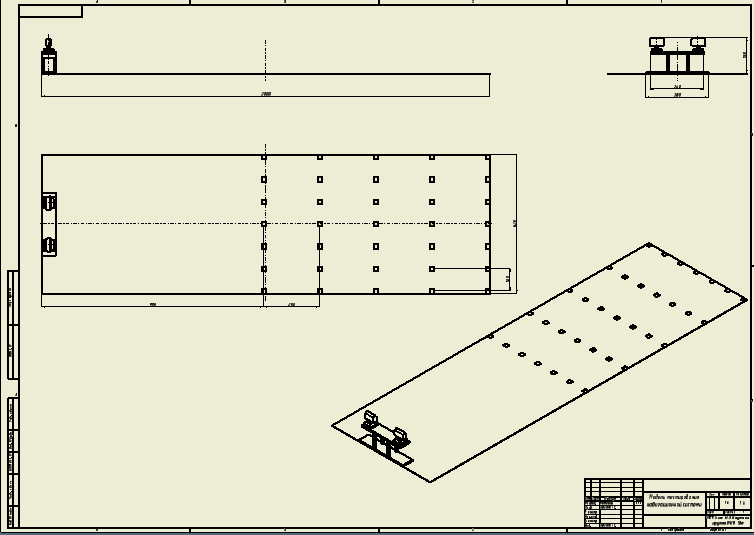
Программная документация состоит из:

1. Техническое задание;
2. Руководство пользователя;
3. Программа и методики испытаний;
   * 1. **Требования к климатическим условиям эксплуатации**

Система предназначена для эксплуатации в комнатном помещении, при температуре от +15º до +35º, с влажностью воздуха не более 50%. Система не предназначена для использования при повышенных температурах или экстремальных условиях.

1. **Концептуальное проектирование**
   1. **Методика тестирования системы**

С целью тестирования разрабатываемого программно-аппаратного устройства была разработана калибровочная модель с использованием системы трехмерного твердотельного моделирования и поверхностного параметрического проектирования (САПР) Autodesk Inventor. Она отображена на рисунке 4.



**стереосистема**

**Позиции установки маркера**

Рис.4 Модель тестирования оптической хирургической навигационной системы

Для проведения тестирования была создана модель, состоящая из миллиметровой бумаги, сферических маркеров (Рисунок 5), стоек разной высоты и стереосистемы.

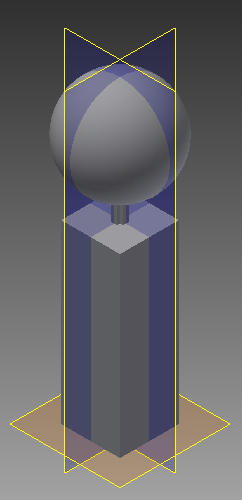


Рис. 5 Модель сферического маркера, для тестирования установки

Перед началом тестирования, пользователь устанавливает стереосистему, закрепленную на стойке, в центр меньшей из сторон миллиметрового листа. После этого происходит калибровка системы, после которой стереосистема не перемещается.

Пользователь устанавливает маркер поочередно в каждую из отметок. После этого система, с заданной частотой кадра, вычисляет координаты маркера в пространстве. После этого, маркер перемещается в следующую точку.

* 1. **Концептуальная модель модуля визуализации**

При проектировании класса визуализации GUI была использована методология функционального моделирования IDEF 0. Ее основные функции были представлены в виде взаимосвязанных функциональных блоков, позволяющих проследить параметры и модули, используемые при проектировании.

Входными параметрами к классу визуализации GUI являются изображения, полученные с правой и левой камеры стереосистемы, а также, начальные значения параметров для поиска карты неравенства и поиска окружностей на изображении. На выходе данного модуля пользователь получает окно, в котором отображаются правое и левое изображения, полученные стереосистемой. Построенная карта неравенств, результат работы детектора Канни, а также отображаются подсчитанные координаты до сферического маркера.

В качестве основного механизма данной модели используется компьютерная система, позволяющая производить вычисления всех требуемых параметров. На этапе тестирования, пользователь вручную устанавливает значения для алгоритма поиска маркеров и вычисления карты глубины. После подбора оптимальных значений, они заносятся в память, и в основном режиме работы используются по умолчанию. На рисунке 6 представлен функциональный блок, отображающий систему. На рисунке 7 изображена его декомпозиция.

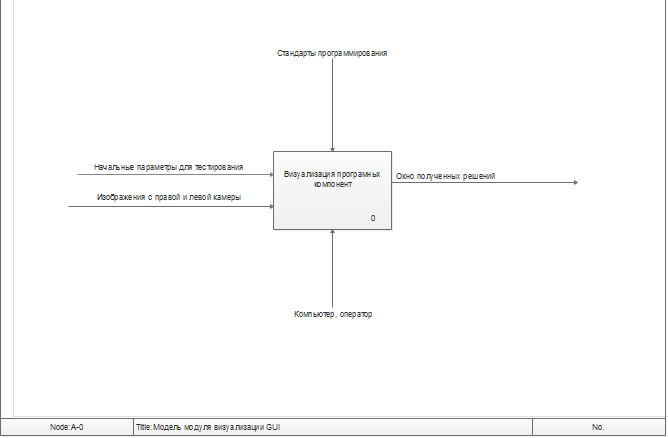


Рис. 6 Верхний уровень модели модуля визуализации GUI

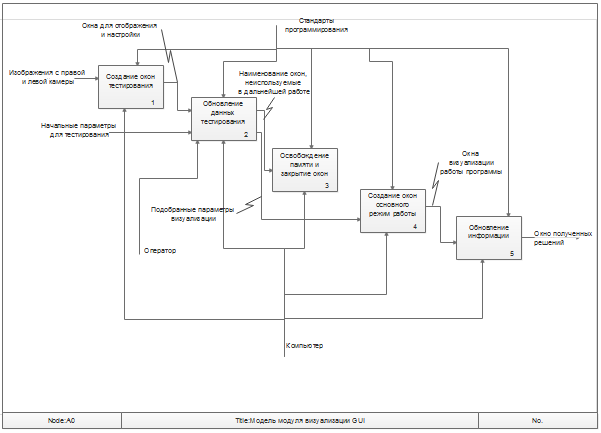
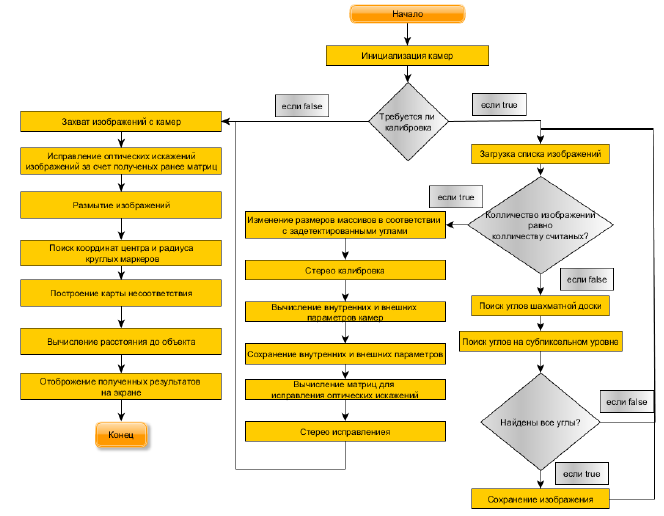


Рис. 7 Декомпозиция верхнего уровня модели

* 1. **Алгоритм разрабатываемой программы**

Перед написанием программной части данного проекта, был разработан алгоритм, представленный на рисунке 8. Программная реализация данного алгоритма представлена в Приложении 2.



1. **Рабочее проектирование системы**
   1. **Библиотека OpenCV**

Библиотека OpenCV - это библиотека компьютерного зрения с открытым исходным кодом разработанная корпорацией Intel, и находящаяся в свободном доступе для скачивания и использования в коммерческих и исследовательских областях. Библиотека написана на Си, Си++ и Python, и запускается под системами Linux, Windows и Mac OSx. OpenCV активно портируется на другие языки, такие как Matlab, C# и другие. Эта библиотека содержит более 500 функций для большинства различных областей в компьютерном зрении, таких как медицинская визуализация, безопасность, калибровка камер, робототехника и компьютерное зрение. Три основных компонента в OpenCV (Рисунок 9), которые используются в исследованиях стереозрения – это компоненты CV, HighGUI и CXCORE. Компонент CV используется для обработки изображения и содержит все алгоритмы компьютерной визуализации, которые используются для обработки стереоизображения. Компонент HighGUI используется для чтения, записи и отображения изображений, а также для создания инструментов-ползунков, используемых для регулировки параметров для методов соответствия. CXCORE содержит структуры данных для изображений и матриц, создаваемых в процессе разработки системы.

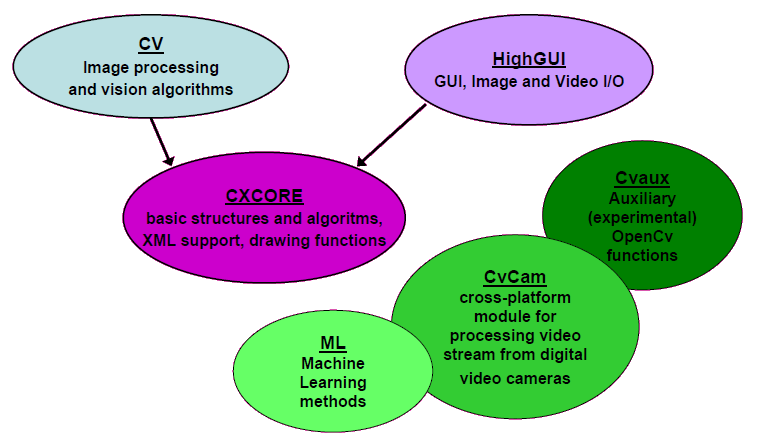


Рис. 9 Основные модули библиотеки OpenCV

Для автоматической сборки библиотеки под операционной системой Windows 10 использовалась программа CMake 3.4.3. CMake – кроссплатформенная утилита, предназначенная для сборки проекта используя исходный код. Чтобы собрать проект для Visual Studio 2015 был написан код представленный в Приложении 1. Он хранит правила и цели сборки.

* 1. **Поиск сферических маркеров**

Поиск объектов в машинном зрении является одной из важнейших задач. В рамках данного курсового проекта был использован метод Хафа для поиска сферических маркеров на плоском изображении. Данный алгоритм позволяет находить объекты, заданные уровнением. В основе данного метода лежит процедура голосования, которая применяется к пространству параметров, из которого и получаются окружности по локальному максимуму и так называемому накопительному пространству, которое строится при вычислении трансформации Хафа.

Точки требуемой окружности можно представить в виде уравнения:

(1)

, где – координаты центра окружности, а R – ее радиус.

Таким образом пространство Хафа будет состоять из трех параметров: координат центра и радиуса. Если пространство Хафа имеет n параметров и эти параметры принимают M дискретных значений, то временная функция сложности будет иметь вид . Поэтому метод поиска окружностей Хафа был преобразован в градиентный метод.

Алгоритм работы метода Хафа представлен на рисунке 10.

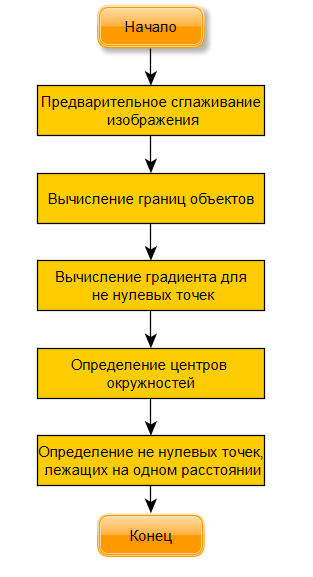


Рис. 10 Алгоритм метода Хафа

Изначально полученное изображение подвергается сглаживанию для того, чтобы избавиться от шума на изображении. Для этого применяется фильтр Гаусса:

После избавления от шума на изображении, на изображении требуется вычислить границы объектов. Для этого воспользуемся детектором границ Кэнни. В его основе лежит следующий алгоритм (Рисунок 11):

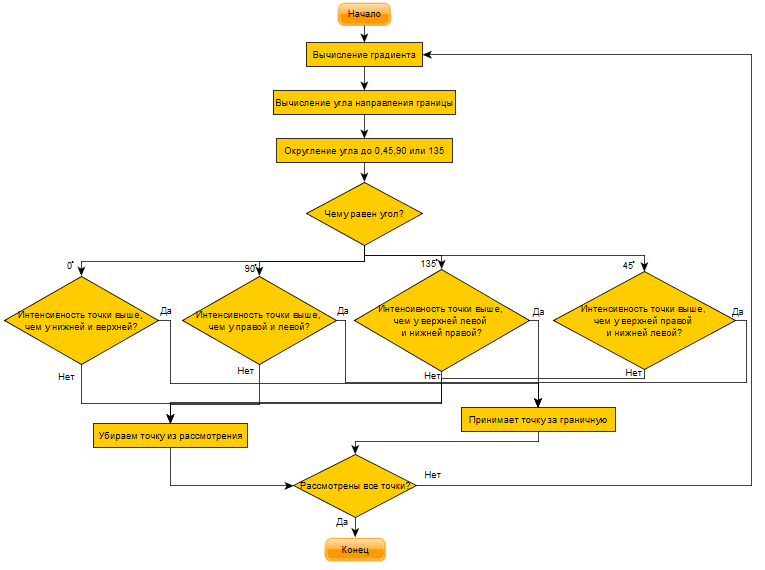


Рис. 11 Алгоритм работы детектора Кэнни

Результат работы детектора Кэнни представлен на рисунке 12.

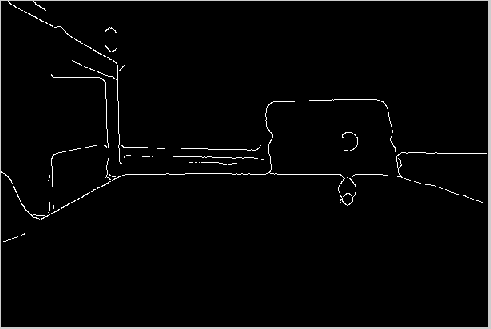


Рис. 12 Границы, полученные при использовании детектора Кэнни

После нахождения границ, производится поиск градиента для не нулевых точек. Представив уравнение 1 в полярных координатах, получим:

(2)

(3)

, где (x0, y0) – координаты центра окружности, R – радиус, а θ – это угол между градиентом и осью *Ox.* Таким образом, избавившись от радиуса, получим уравнение для координат центра окружности:

(4)

Подставляя в формулу 4 координаты не нулевых пикселей с соответствующим углом, можно получить массив координат (x0, y0) вдоль линии. Локальный максимум в данном массиве даст нам центр окружности. После этого для отдельного центра определяются не нулевые точки, лежащие на одном расстоянии.

В библиотеке OpenCV, представленный алгоритм реализован в следующей функции:

*void HoughCircles (InputArray image,*

*OutputArray circles,*

*int method,*

*double dp,*

*double minDist,*

*double param1=100,*

*double param2=100,*

*int minRadius=0,*

*int max Radius=0);*

Данная функция принимает девять параметров, а возвращает один. Первый параметр – это входное изображение, на котором будет проводиться поиск окружностей. Найденные на изображении окружности записываются в массив векторов *circles*, каждый из которых состоит из трех элементов: координата x, координата y и радиус окружности. Переменная *method* имеет только одно значение CV\_HOUGH\_GRADIENT, в она передает какой метод используется при поиске. Dp – это переменная задающая обратное разрешение массива хранящего центры окружностей, чем у исходного изображения. Если dp=2, то массив, хранящий центры будет в два раза больше. minDist – минимальное расстояние между центрами двух окружностей. Param1 и Param2 значения порога для построения карты границ методом Канни, по умолчанию они имеют значения равные 100. Последние два параметра это минимальный и максимальный радиус детектируемых окружностей. Результата работы данной функции представлен на рисунке 13.

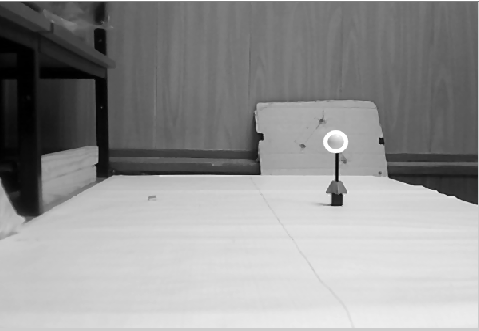


Рис. 13 Результат работы функции *HoughCircles*

* 1. **Получение карты неравенства и расчет координат**

Чтобы определить расстояние от камеры, должно быть найдено несоответствие, изменение расположения точек в левом изображении относительно правого. Из этого следует, что должно быть перекрытие на двух изображениях, так чтобы точка на левом изображении, также существовала на правом, и можно было найти соответствие.

Алгоритм, который был использован мной для вычисления корреляции и поиска несоответствия – это метод сопоставления блоков, где окна суммы абсолютных расстояний (SAD) используются для нахождения соответствий. Окна SAD используются, как метод измерения соответствия для каждого пикселя на изображение на основе окружающих его соседей (Рисунок 14). Есть три шага для метода сопоставления блоков, которые использует OpenCV; пред-фильтрация, поиск соответствий и пост-фильтрация.

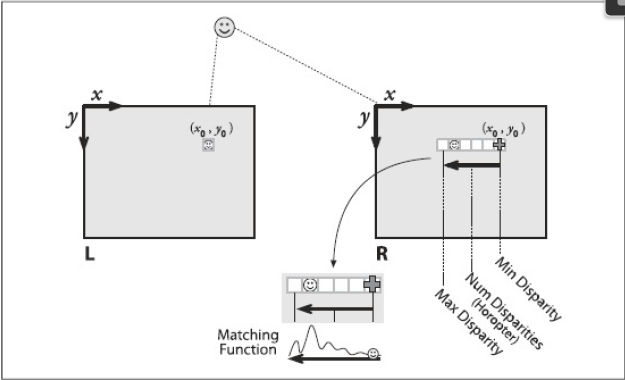


Рис. 14 Иллюстрация принципа работы метода сопоставления блоков (Bradski G., 2008)

После устранения искажений, строки на изображениях выравниваются таким образом, чтобы соответствующие точки лежали на одной и той же горизонтальной прямой в обоих изображениях (Рисунок 15).

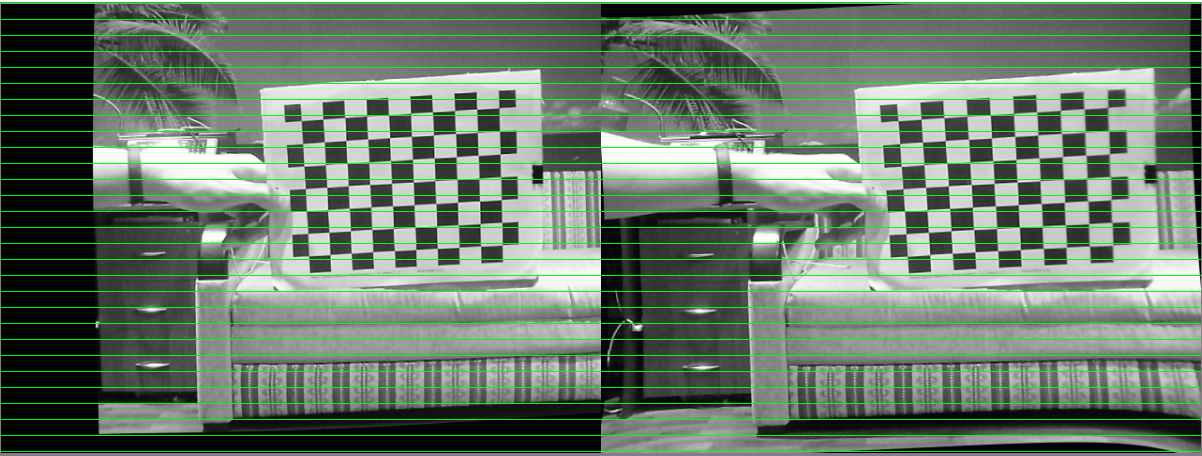


Рис. 15 Исправленные изображения

Окна SAD устанавливаются на пиксель в левом изображении и рассчитывается оценка. Затем алгоритм ищет соответствующий пиксель на правом изображение, начиная с той же координаты, что и на левом изображении, и движется влево, вдоль оси X, вычисляя оценку для каждого положения пикселя до тех пор, пока она достигает максимального неравенства. Неравенство – это количество пикселей смещения от исходного пикселя до пикселя, на котором алгоритм сопоставления закончил работать. Это значение SAD вычисляется, используя следующее уравнение:

(5)

В этом уравнении, (r,c) – это координаты точки, которые ищутся для сопоставления в правом изображении, d – это несоответствие точки на правом изображении от оригинальной точки и w – размер окна, которое помещается вокруг каждой точкой. Это уравнение показывает, что оценки вычисляются, основываясь на значение интенсивности соседних пикселей, окружающих точку. Точка на правом изображении, с минимальным значением внутри области поиска, лучшим сопоставлением для точки на левом изображении. Точки с большим значением несоответствия представляют точки, которые ближе к камере, а маленькое несоответствие показывает, что они находятся дальше от камеры.

Пост фильтрация выполняется для удаления шумов, которые содержат ложные соответствия. Для этого, OpenCV использует уникальность соотношений, а также порог текстуры. Уникальность соотношений используется, чтобы удостовериться, что значение, которое было вычислено для сопоставленной точки, не является самой близкой отметкой, но является посторонним значением, где она окружена десятками, которые далеки от сопоставления. Порог текстуры устанавливается таким образом, что шум может быть уменьшен во время процесса согласования, без оценки, которая считается ниже порогового значения.

Результат метода совпадений – это изображение где, каждый пиксель является неравенством, которое было найдено из левого и правого изображения, это изображение называется картой неравенства. Примером карты неравенства из оригинального изображения показано на Рисунке 16. Яркие значения интенсивности показывают объекты, которые находятся ближе к камере, более темные объекты – это те, которые находятся дальше от камеры. Черные пиксели – это те точки, где никакого соответствия не было найдено между изображениями.

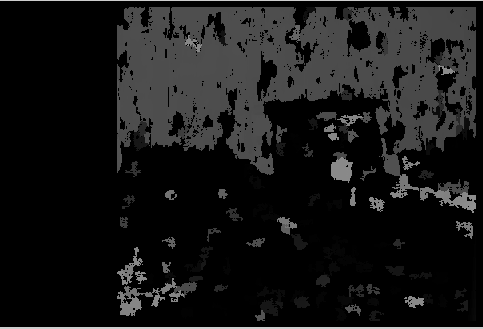


Рис. 16 Полученная карта глубины

Перепроектирование двумерных точек из набора изображений в трехмерные точки выполняется, используя значения разности каждого пикселя. Каждый пиксель имеет соответствующие (X, Y, Z) координаты, которые характеризуют его позицию в трехмерном пространстве. Z координата находится используя уравнение, связывающее расстояние в зависимости от базового уровня, фокусного расстояния и несоответствия, которое показано ниже:

(6)

В этой формуле, значение несоответствия находится при помощи перемножения корреляционной процедуры на текущий размер отдельного элемента на датчике изображения, обеспечивая неравенство, как показатель расстояния. Значения X и Y находятся при помощи поиска размера пикселя из их расчетного состояния, найденного для Z. Во-первых, поле зрение (Рисунок 17) должно быть известно или вычислено используя следующее уравнение:

(7)

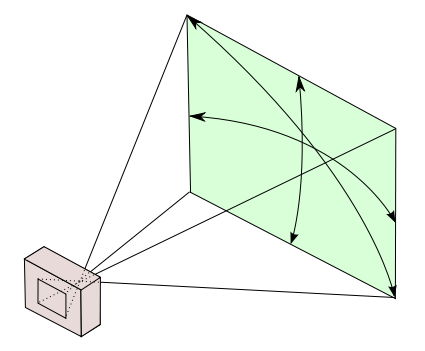


Рис. 17 Горизонтальное, вертикальное и диагональное поле зрения камеры

Размер значения сенсора – это общий размер по вертикали для вычисления вертикальных FOV и горизонтальный размер для горизонтальных FOV. После того, как вычисляется FOV, размер пикселя может быть найден на заданном расстоянии, используя следующее уравнение:

(8)

Используя горизонтальные и вертикальные FOV будут давать горизонтальное и вертикальное расстояние, охваченное изображением. Исходя из этого измерения и деления его на общее числи горизонтальных или вертикальных пикселей, обеспечивается значение расстояния, пройденного в пиксель на расстоянии Z от камеры. В итоге, используя координаты центра из левой камеры, cx и cy - эти X и Y координаты реального мира могут быть определены координаты пикселей x и y, используя (cx, cy) как центр проекции. Следующие уравнения показывают, как вычисляются x и y:

(9)

(10)

Используя формулы 9 и 10, облако точек может быть легко пере проецировано в любую систему измерения, требуемую от координат пикселя и их соответствующих значений неравенства.

* 1. **Визуализация полученных данных**

Для улучшения читаемости кода и повышения наглядности работы программы, был разработан класс GUI, представленный в Приложении 3. Разработка кода была реализована в соответствии с составленной моделью IDEF0 в главе 3.2. Результатом работы класса является окно настроек (Рисунок 18) и окно визуализации полученной информации (Рисунок 19). В окне настроек производится подбор оптимальных параметров для функций поиска сферических маркеров и построения карты неравенств. В окне визуализации полученной информации представлены изображения с левой и правой камеры, вычисленная карта неравенств и результат работы детектора Кэнни. Также здесь отображаются вычисленные значения координат маркера.

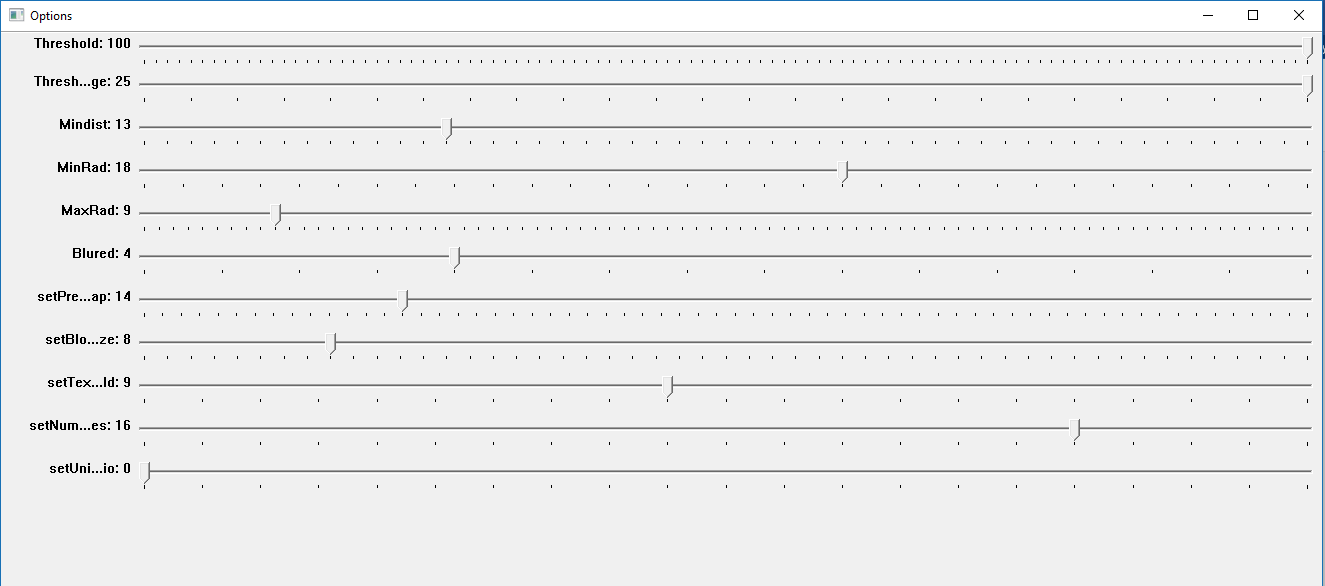


Рис. 18 Окно настроек

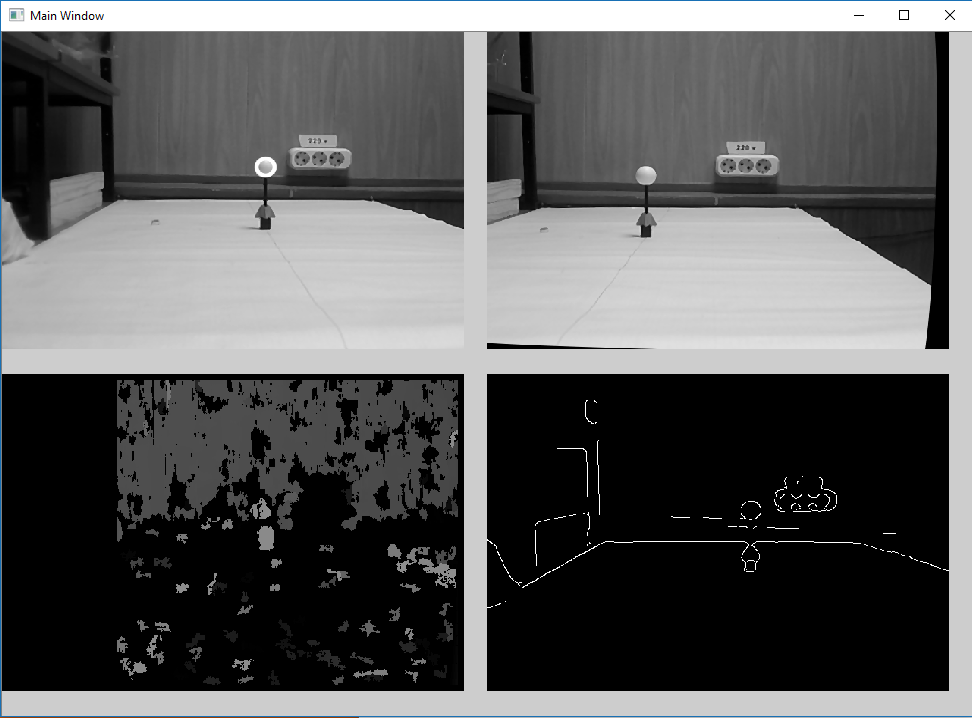


Рис. 19 Окно визуализации полученной информации

# **ЗАКЛЮЧЕНИЕ**

В данном курсовом проекте, был проведен анализ существующих решений в хирургической навигации, представлены виды и характеристики навигационных систем. В соответствии с предпроектным исследование, было решено разрабатывать оптическую хирургическую навигационную систему, а также было представлено техническое задание на разработку данного программно-аппаратного комплекса. При помощи системы автоматического проектирования (САПР) Autodesk Inventor 2016 была смоделирована модель установки для тестирования, разрабатываемого программного обеспечения. Был разработан алгоритм программы, выполняющей поиск сферических маркеров, вычисление карты неравенства и расчет координат маркера, а также при помощи методологии функционального моделирования IDEF0 была построена модель класса визуализации GUI. Используя кроссплатформенную утилиту CMake 3.4.3 было собрано решение для реализации проекта на языке C++. Используя библиотеку OpenCV была разработана программа, позволяющая производить поиск сферических маркеров, расчет карты неравенств и подсчет координат, задетектированных маркеров. Также, был реализован класс визуализации GUI. В дальнейшем планируется совершенствование данного программно-аппаратного комплекса и повышение точности детектирования.

# **СПИСОК ЛИТЕРАТУРЫ:**

# 

1. Bradski G., K. A. *Learning OpenCV: Computer Vision with the OpenCV Library.* O'Reilly. (2008)
2. R. Khadem, C. Y.-T. Comparative traking error analysis of five different optical tracking system. *Comput Aided Surg*, 98-107. (2000)
3. Yuen, H. K. and Princen, J. and Illingworth, J. and Kittler, J., Comparative study of Hough transform methods for circle finding. Image Vision Comput. 8 1, pp 71–77 (1990)
4. Janne Heikkilä and Olli Silvén. A Four-step Camera Calibration Procedure with Implicit Image Correction. Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on. –P. 1106-1112. (1997)
5. Koivukangas, Tapani. Methods for determination of the accuracy of sergical guidance devices. A study in the region of neurosurgical interest, P.O. Box 5000, FI-90014 University of Ouly, Finland. Acta Univ. Oul. C. 427, 2012.
6. Тымкович М. Ю. Оптические методы регистрации пространственного положения хирургического инструмента в компьютерной навигационной системе. Вiсник НТУ "XПI". c 124-130. (2012)

# **ПРИЛОЖЕНИЕ 1**

**cmake\_minimum\_required(VERSION 3.3)**

**#This is needed to set OpenCV libs properly**

**#message (STATUS "CMAKE\_MODULE\_PATH=${CMAKE\_MODULE\_PATH}")**

**set(CMAKE\_MODULE\_PATH ${CMAKE\_MODULE\_PATH} ${CMAKE\_SOURCE\_DIR})**

**message (STATUS "CMAKE\_MODULE\_PATH=${CMAKE\_MODULE\_PATH}")**

**project (SurgeryNavigation)**

**set (CMAKE\_BUILD\_TYPE Debug)**

**message (STATUS "Provided build type: ${CMAKE\_BUILD\_TYPE}")**

**if(NOT DEFINED CMAKE\_BUILD\_TYPE)**

**set (CMAKE\_BUILD\_TYPE CACHE STRING "Build Type" FORCE)**

**message (FATAL\_ERROR "No build type provided! (${CMAKE\_BUILD\_TYPE})")**

**endif()**

**# using CMAKE\_BUILD\_TYPE variable passed through the command line**

**if (CMAKE\_BUILD\_TYPE STREQUAL "Debug")**

**set (OpenCV\_DIR "C:/dev/OpenCV/install.debug")**

**set (CMAKE\_CONFIGURATION\_TYPES "Debug" CACHE STRING "Build Type" FORCE)**

**endif()**

**if(CMAKE\_BUILD\_TYPE STREQUAL "Release")**

**set (OpenCV\_DIR "C:/dev/OpenCV/install.release" CACHE STRING "Build Type" FORCE)**

**set (CMAKE\_CONFIGURATION\_TYPES "Release" CACHE STRING "Build Type" FORCE)**

**endif ()**

**message (STATUS "Configuring for: ${CMAKE\_BUILD\_TYPE}")**

**message (STATUS "Looking for OpenCV at: ${OpenCV\_DIR}")**

**message(STATUS "opencv\_PATH=${opencv\_PATH}")**

**set(OpenCV\_LIBS\_USED core calib3d video highgui features2d flann)**

**find\_package(OpenCV REQUIRED NO\_MODULE COMPONENTS ${OpenCV\_LIBS\_USED} PATHS ${OPENCV\_DIR} NO\_DEFAULT\_PATH)**

**foreach (MODULE ${OpenCV\_LIBS})**

**get\_target\_property(PROP ${MODULE} LOCATION\_${CMAKE\_BUILD\_TYPE})**

**get\_filename\_component(FILENAME ${PROP} NAME\_WE)**

**set (NEW\_NAME "${OpenCV\_LIB\_PATH}/${FILENAME}.lib")**

**list(APPEND OPENCV\_LIBS\_WITH\_PATHS ${NEW\_NAME})**

**message ("FILENAME="${FILENAME})**

**message ("NEW\_NAME="${NEW\_NAME})**

**endforeach()**

**message(STATUS "OpenCV\_FOUND=${OpenCV\_FOUND}")**

**message(STATUS "OpenCV\_LIB\_COMPONENTS=${OpenCV\_LIB\_COMPONENTS}")**

**message(STATUS "OpenCV\_LIBS=${OpenCV\_LIBS}")**

**message(STATUS "\_OpenCV\_LIB\_PATH=${\_OpenCV\_LIB\_PATH}")**

**include\_directories(${OpenCV\_INCLUDE\_DIRS})**

**link\_libraries(${OpenCV\_LIBS})**

**message(STATUS "OpenCV library status:")**

**message(STATUS " version: ${OpenCV\_VERSION}")**

**message(STATUS " libraries: ${OpenCV\_LIBS}")**

**message(STATUS " include path: ${OpenCV\_INCLUDE\_DIRS}")**

**add\_executable(Capture\_Video main.cpp GUI.cpp)**

**message (STATUS "List of files for linker input: ${OPENCV\_LIBS\_WITH\_PATHS}")**

**target\_link\_libraries(Capture\_Video ${OPENCV\_LIBS})**