

ПЕРВОЕ ВЫСШЕЕ ТЕХНИЧЕСКОЕ УЧЕБНОЕ ЗАВЕДЕНИЕ РОССИИ



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
САНКТ-ПЕТЕРБУРГСКИЙ ГОРНЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ

Руководитель ОПОП ВО

Профессор В.Ю. Бажин

УЧЕБНО-МЕТОДИЧЕСКИЕ РАЗРАБОТКИ
для проведения практических занятий по учебной дисциплине

**«СОВРЕМЕННЫЕ СИСТЕМЫ КОНТРОЛЯ В
МЕТАЛЛУРГИИ»**

Системы технического зрения

Направление подготовки: *09.06.01 Информатика и вычислительная техника*

Программы:

Автоматизация и управление технологическими процессами и производствами (металлургия)

Автоматизация и управление технологическими процессами и производствами (химическая промышленность)

Разработал: *доцент Кульчицкий А.А., ст. препод. Булатов В.В.*

САНКТ-ПЕТЕРБУРГ

УДК 53.087.92, 621.317.39.084.2

СОВРЕМЕННЫЕ СИСТЕМЫ КОНТРОЛЯ В МЕТАЛЛУРГИИ (системы технического зрения): Методические указания к практическим занятиям / Санкт-Петербургский горный университет . Сост.: *А.А. Кульчицкий. В.В. Булатов* СПб, 2018, 43 с.

Методические указания к практическим занятиям разработаны в соответствии с требованиями федеральных государственных образовательных стандартов высшего образования. Изложен порядок выполнения практических занятия, посвященных системам технического зрения. Данные методические указания предназначены для магистрантов очной формы обучения направления 09.06.01 Информатика и вычислительная техника, по программам «Автоматизация и управление технологическими процессами и производствами (металлургия)» и «Автоматизация и управление технологическими процессами и производствами (химическая промышленность)»

Научный руководитель проф. *Бажин В.Ю.*

© Санкт-Петербургский горный университет, 2018 г.

Методика разработки систем технического зрения

1. Программное обеспечение

1.1. NI Vision Builder AI v.3.6

Для разработки программы распознавания геометрических образов выбрана среда **NI Vision Builder AI v.3.6**.

NI Vision Builder for Automated Inspection представляет собой настраиваемое ПО для систем машинного зрения, предназначенное для моделирования, испытаний и внедрения приложений. Работа в NI Vision Builder for Automated Inspection осуществляется без программирования, однако позволяет реализовывать мощные приложения на уровне возможностей LabVIEW. Программа содержит встроенные средства сбора и анализа изображений и решения задач идентификации и сортировки. Кроме того, имеются возможности определения соответствия объекта шаблону, управления цифровыми линиями ввода/вывода и взаимодействия с внешними устройствами, например, промышленными контроллерами. Основные функции:

- Интерактивная оболочка разработки приложений машинного зрения
- Автоматическая генерация кодов LabVIEW
- Управление цифровым вводом/выводом, ПЛК и другими устройствами через последовательный порт
- Взаимодействие с устройствами ввода изображений компании NI для PCI, PXI и промышленными FireWire камерами.

Программный продукт NI Vision builder AI и интеллектуальная камера NI 17xx позволяют автоматизировать визуальный контроль на предприятии, исключив человека из процесса контроля, а также повысить скорость и точность операций контроля. NI Vision builder позволяет управлять интеллектуальной камерой, внешним источником света (вспышкой), а также различным промышленным оборудованием по протоколу Modbus, TCP/IP и serial ports.

Данная программная среда не требует освоения языка программирования и позволяет реализовать алгоритмы различной сложности с помощью блок-схем и несложных для понимания функций.

NI Vision builder AI позволяет получать и обрабатывать изображения непосредственно с интеллектуальной камеры и работать с изображениями,



расположенными на локальном диске компьютера или сервера. Программа работает в двух режимах:

- режим конфигурации (configure inspection) позволяет создавать новые и изменять уже созданные алгоритмы обработки изображений. Этот режим позволяет использовать средства отладки алгоритмов.
- режим инспекции (inspect product) исполняет выбранный алгоритм и отображает результаты его работы.

1.2. Labview.

Графический язык программирования «G», используемый в Labview, основан на архитектуре потоков данных. Последовательность выполнения операторов в таких языках определяется не порядком их следования (как в императивных языках программирования), а наличием данных на входах этих операторов. Операторы, не связанные по данным, выполняются параллельно в произвольном порядке.

В основе программирования в LabVIEW лежит понятие Виртуальных приборов (Virtual Instruments, VI). На лицевой панели, как и положено, располагаются элементы управления программой – кнопки, графики, выключатели и тому подобное. Блок-схема – это, по сути, и есть сама программа. При написании (а вернее создании, потому что писать приходится не так уж и много) программы используется такое понятие, как «поток данных» (Data Flow). Суть его в том, что все элементы программы (которые представлены графически) связываются между собой связями (проводами, нитками) по которым и происходит передача данных. В общем, описать это довольно сложно, лучше посмотреть на картинку, рис. 3.

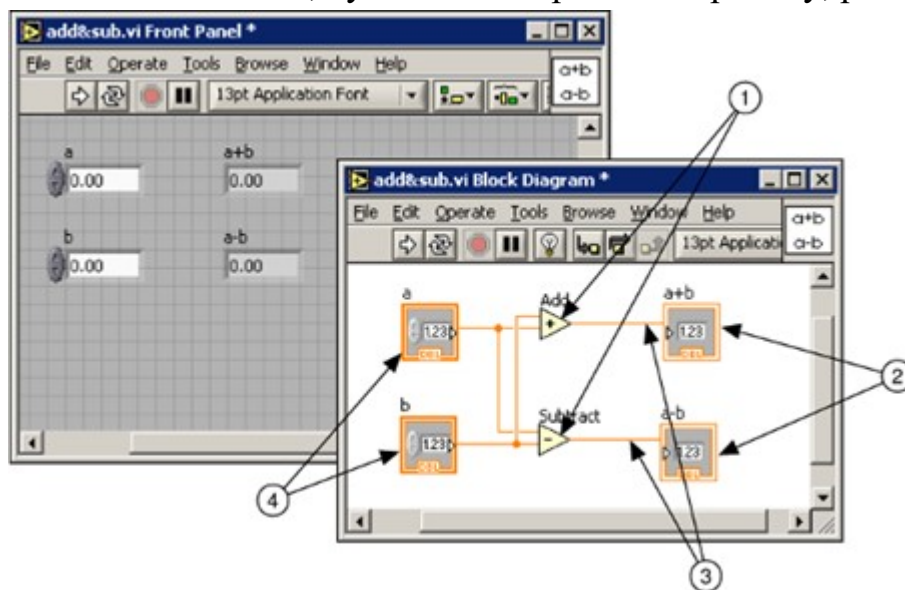


Рис. 1. Простейший прибор.

Цифрами обозначены:

- Точки, элементы программы (Nodes)

- Терминалы индикаторов (Indicator Terminals)
- Связи (Wires)
- Терминалы управляющих элементов (Control Terminals)

Итак, в LabVIEW вы создаете пользовательский интерфейс (лицевую панель), с управляющими элементами и индикаторами. Управляющие элементы – это тумблеры, кнопки, поля ввода и прочие устройства ввода. Индикаторы – это графики, шкалы, лампочки, текстовые поля и тому подобное. После создания пользовательского интерфейса, вы добавляете программный код, который управляет объектами на лицевой панели. Этот код содержится в схеме (block diagram). Этот код чем-то напоминает собой блок-схему, хотя отличий много.

LabVIEW можно использовать для того, чтобы управлять различным оборудованием, таким, как, устройства сбора данных, различные датчики, устройства наблюдения, двигательные устройства (например, шаговые моторы) и тому подобное, а так же GPIB, PXI, VXI, RS-232 и RS-484 устройства. Также в LabVIEW имеются встроенные средства для подключения созданных программ к сети, используя LabVIEW Web Server и различные стандартные протоколы и средства, такие как TCP/IP и ActiveX.

Используя LabVIEW, можно создавать приложения для тестирования и измерений, сбора данных, управления различными внешними устройствами, генерации отчетов. Так же можно создать независимые исполняемые файлы и библиотеки функций, такие как DLL, так как LabVIEW – это полноценный 32-битный компилятор.

Достоинствами этого ПО являются:

- Полноценный язык программирования
- Интуитивно понятный процесс графического программирования
- Широкие возможности сбора, обработки и анализа данных, управления приборами, генерации отчетов и обмена данных через сетевые интерфейсы
- Драйверная поддержка более 2000 приборов
- Возможности интерактивной генерации кода
- Шаблоны приложений, тысячи примеров
- Высокая скорость выполнения откомпилированных программ
- Совместимость с операционными системами Windows2000/NT/XP, Mac OS X, Linux и Solaris.

LabVIEW поддерживает огромный спектр оборудования различных производителей и имеет в своём составе (либо позволяет добавлять к базовому пакету) многочисленные библиотеки компонентов:

- для подключения внешнего оборудования по наиболее распространённым интерфейсам и протоколам (RS-232, GPIB 488, TCP/IP и пр.);
- для удалённого управления ходом эксперимента;
- для управления роботами и системами машинного зрения;
- для генерации и цифровой обработки сигналов;
- для применения разнообразных математических методов обработки данных;
- для визуализации данных и результатов их обработки (включая 3D-модели);
- для моделирования сложных систем;
- для хранения информации в базах данных и генерации отчетов;
- для взаимодействия с другими приложениями в рамках концепции COM/DCOM/OLE и пр.

Вместе с тем LabVIEW – очень простая и интуитивно понятная система. Неискушённый пользователь, не являясь программистом, за сравнительно короткое время (от нескольких минут до нескольких часов) способен создать сложную программу для сбора данных и управления объектами, обладающую красивым и удобным человеко-машинным интерфейсом. Например, средствами LabVIEW можно быстро превратить старый компьютер, снабжённый звуковой картой, в мощную измерительную лабораторию.

Специальный компонент LabVIEW – Application Builder, позволяет выполнять LabVIEW-программы на тех компьютерах, на которых не установлена полная среда разработки.

LabVIEW находит применение в самых разнообразных сферах человеческой деятельности. В соответствии со своим названием он первоначально использовался в исследовательских лабораториях, да и в настоящее время является наиболее популярным программным пакетом как в лабораториях фундаментальной науки (например, Lawrence Livermore, Argonne, Batelle, Sandia, Jet Propulsion Laboratory, White Sands и Oak Ridge в США, CERN в Европе), так и в отраслевых промышленных лабораториях. Все более широкое применение LabVIEW находит в образовании, в университетских лабораторных практикумах – особенно по предметам электротехники, механики и физики.

Распространение LabVIEW за пределами лабораторий пошло по всем направлениям: вверх (на борту космических аппаратов), вниз (на подводных лодках) и по горизонтали (от буровых установок в Северном море до промышленных предприятий в Новой Зеландии). В связи с ростом

возможностей Internet сфера применения LabVIEW стала расширяться не только в географическом, но и в виртуальном пространстве (cyberspace). Все большее число разработчиков создает виртуальные приборы, допускающие удаленное управление и наблюдение через Internet. Измерительные системы на основе виртуальных приборов отличаются своей многофункциональностью, гибкостью и низкой стоимостью как с точки зрения оборудования, так и с точки зрения затрат времени на разработку.

В настоящее время LabVIEW широко применяется в следующих сферах:

- Автомобильная промышленность
- Телекоммуникации
- Аэрокосмическая промышленность
- Полупроводниковая промышленность
- Разработка и производство электроники
- Управлений технологическими процессами
- Биомедицина

Благодаря своей гибкости и масштабируемости, LabVIEW может использоваться на всех этапах технологического процесса: от моделирования и разработки прототипов продуктов до широкомасштабных производственных испытаний. Применение интегрированной среды LabVIEW для измерения сигналов, обработки результатов и обмена данными повысит производительность всего предприятия.

Лучшим способом объяснить причины столь широкого распространения пакета LabVIEW будет обобщение способов его использования. Во всех видах человеческой деятельности существуют области, где не обойтись без определенных видов измерений – очень часто это температурные измерения, например в печах, холодильниках, парниках, технологических помещениях и даже в кастрюле с супом. Кроме температуры, часто измеряют давление, силу, пространственное смещение, механическое напряжение, pH и т.д. – список огромный! Сейчас персональные компьютеры проникли практически во все сферы жизнедеятельности. LabVIEW ускоряет внедрение компьютера в измерительные системы – и не только потому, что облегчает проведение измерений, он также дает возможность проанализировать измеренные величины, отобразить их на графиках и в отчетах и при желании опубликовать. Из выше сказанного можно сделать вывод о всеобъемлющем применении данного способа программирования, он не только позволяет нам измерять величины но и отображать незамедлительно результаты этих измерений без дополнительного программного обеспечения. После измерения и анализа какой-либо величины следующим логическим шагом часто является управление, то есть изменение определенных параметров в

зависимости от полученных результатов. Например, измерив температуру объекта, можно включить устройство для его охлаждения либо нагрева. И вновь LabVIEW значительно облегчает решение этой задачи: *мониторинг* и *управление процессами* являются основными функциями этого программного продукта. Управление процессами может быть прямым или осуществляется через специальные *программируемые логические контроллеры* (programmable logical controllers – PLC), что принято называть *диспетчерским управлением и сбором данных* (supervisory control and data acquisition – SCADA).

2. Техническое обеспечение

2.1. Оптическая скамья

Лабораторный стенд разработан на базе оптической скамьи фирмы «NEW EDUCATION LABORATORY (NINGBO)». Скамья оптическая предназначена для проведения в лабораторных условиях опытов по оптике. Представляет собой массивную направляющую с насаженными на нее штативами (рейтерами) с оптическими элементами. Рейтеры можно перемещать вдоль направляющей и неподвижно закреплять в любом месте по всей длине направляющей. Механические перемещения рейтеров строго параллельны оптическим и визирным осям.



Рис.2 Оптическая скамья

- направляющая с сантиметровыми делениями
- осветитель (лампа накаливания)
- рейтер - 3 шт.
- сферическая короткофокусная собирающая линза
- сферическая длиннофокусная собирающая линза
- экран с диафрагмой
- экран со стрелками
- экран для наблюдения изображений
- цветовой диск

2.2. Камера технического зрения (КТЗ)

Для разработки лабораторного стенда была выбрана камера технического зрения NI 1722

Смарт-камера NI		NI 1722
Процессор	Главный процессор	400 МГц Power PC
	Сопроцессор	-
Память	Системная память	128 МБ
	Постоянная память	128 МБ
Матрица	Разрешение	640x480
	Размер матрицы	1/3 CCD
	Разрядность АЦП	8 бит
	Скорость получения изображения	60 кадров в секунду
	Возможность частичного захвата изображения	Да
Ввод-вывод	TTL I/O	1
	Изолированный цифровой ввод	2
	Изолированный цифровой выход	2
	Подключение энкодера	Нет
	Поддержка Ethernet	Да
Коммуникационные интерфейсы	Ethernet	2 x 10/100/1000
	RS232	Да
Управление подсветкой	Выход на внешний контроллер управления подсветкой	Да
	Встроенный контроллер управления подсветкой	Нет
Средства разработки	Среда разработки, поставляемая с камерой	NI Vision Builder AI
	Дополнительные средства разработки	LabVIEW
Условия эксплуатации	Смарт-камера	0..45 C ⁰
Материал корпуса		Металл

Камера NI 1722 имеет несколько режимов получения изображения, при этом есть возможность выбирать качество полученных снимков и скорости получения изображений:

1. Режим полного сканирования – 60 кадр/с.
2. Режим частичного сканирования 1/2 – 109 кадр/с.
3. Режим частичного сканирования 1/4 – 175 кадр/с.

Вы можете создавать готовые приложения для камер NI с помощью NI LabVIEW Real-Time и модуля NI Vision. Конфигурирование и настройка камер осуществляется в программном пакете NI Vision Builder for Automated Inspection, который входит в комплект поставки.



Рис.3 Камера технического зрения NI 1722

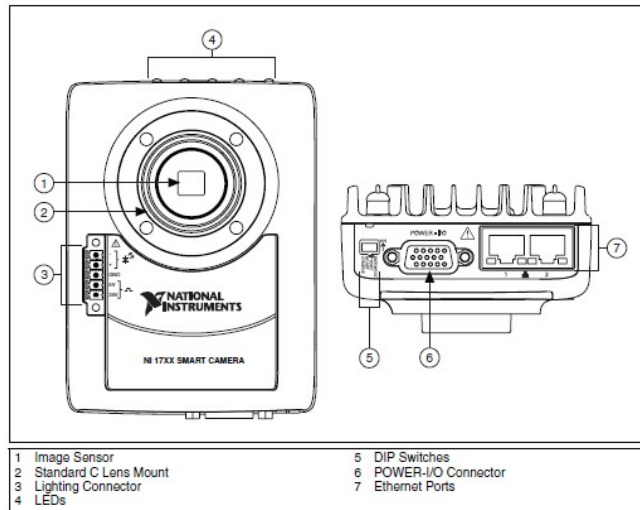
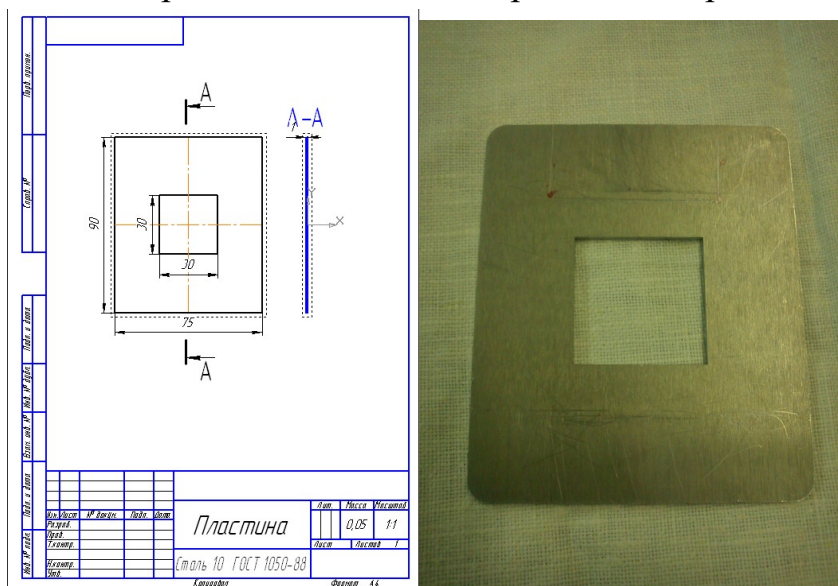


Рис.4 Интеллектуальная камера NI 1722

3. Образцы контроля

В ходе выполнения дипломного проекта были разработаны несколько пластин-образцов для распознавания геометрических образов:



а)

б)

Рис.1. Образец-пластина (квадрат): а)чертеж б)общий вид

2.6 Блок-схема алгоритма работы со стендом.

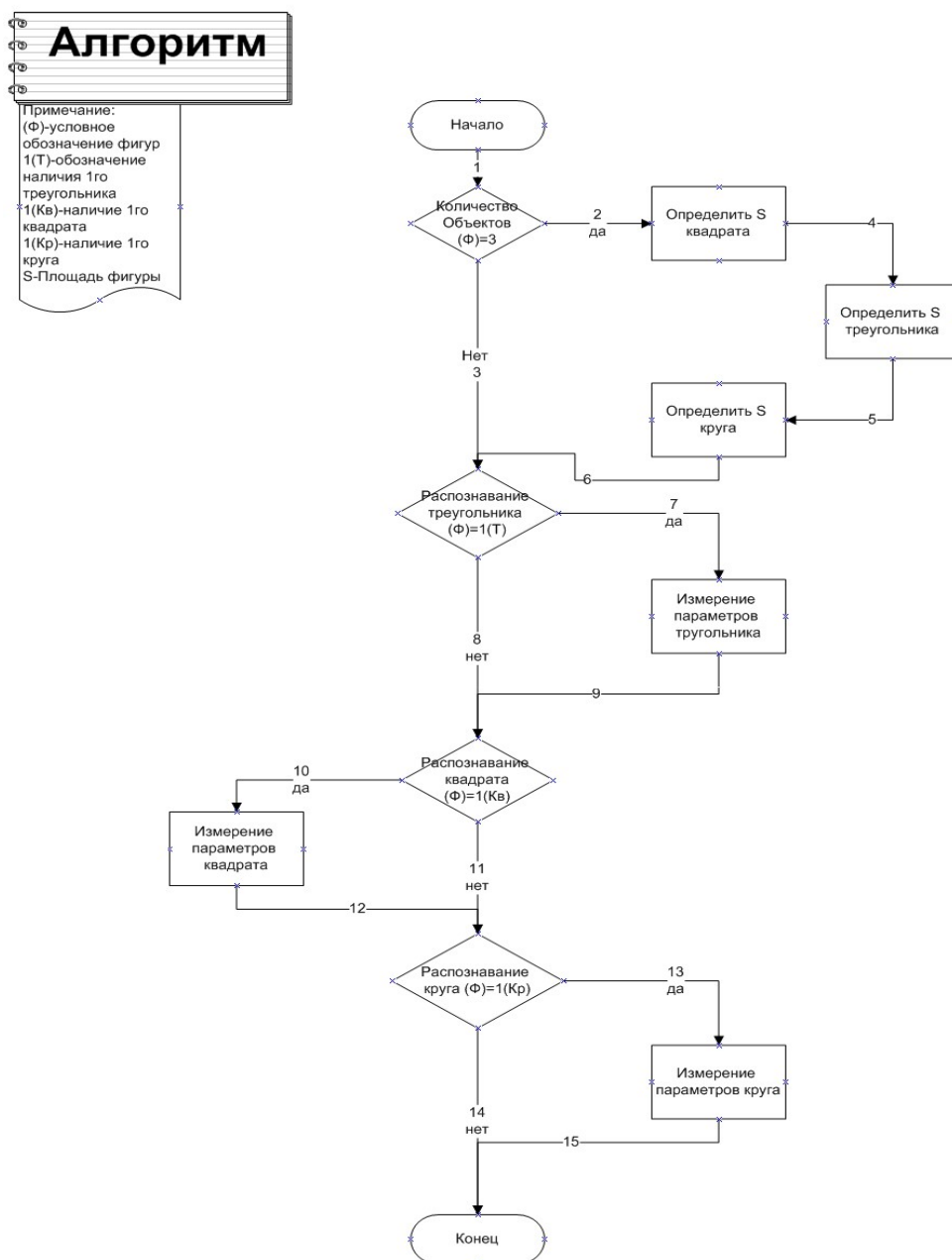


Рис.7. Алгоритм управляющей программы

4. Разработка управляющей программы.

Данный алгоритм, представлен в программе NI Vision Builder, изображен на рисунке 8. Каждый блок содержит несколько последовательно выполняющихся функций, результаты которых заносятся в память программы и используются в следующих функциях. Блоки соединены между собой линиями связи. Переход по связи «default» не имеет условия перехода и выполняется, если нет других связей, удовлетворяющих заданному условию.

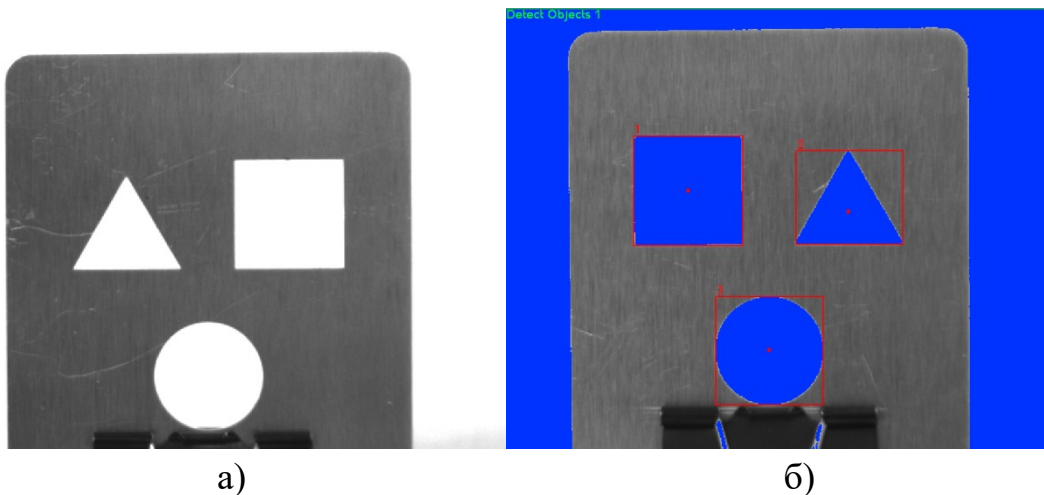


Рис.9 Получение изображения и подсчет объектов а) Изображение, б) Подсчет

Управлять тем, какие объекты будут подсчитаны можно при помощи выбора минимального размера отображаемых объектов в настройках функции Detect Objects

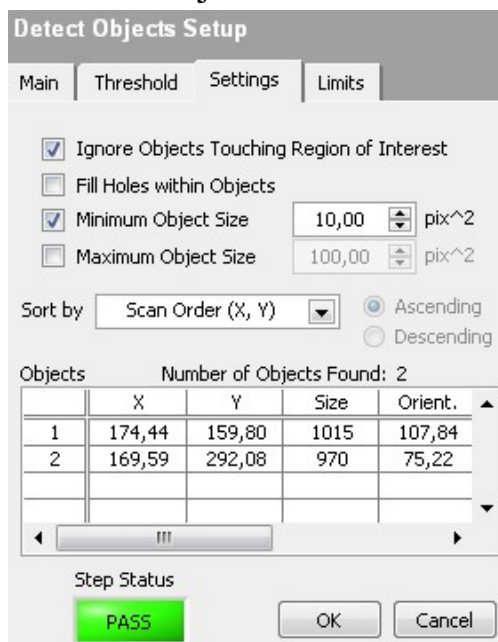


Рис.10 Окно настройки параметров функции

2.1 В зависимости от результата, полученного при подсчете объектов, алгоритм или перейдет к вычислению параметров образца с тремя объектами или же начнет поиск одного из возможных объектов, если количество объектов будет равным трем, то пойдет запуск поочередного определения каждого из объектов на образце одновременно с нахождением их размеров:

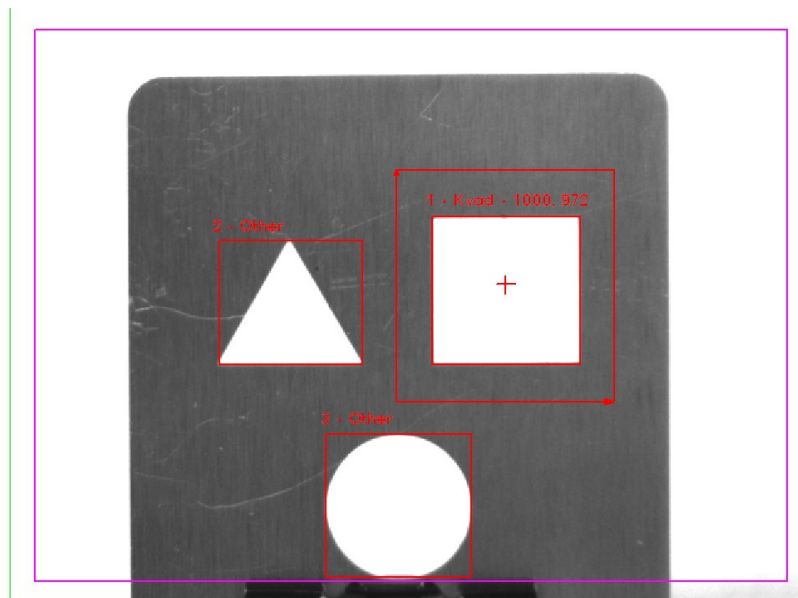


Рис.11 Визуальное отображение распознавания объекта

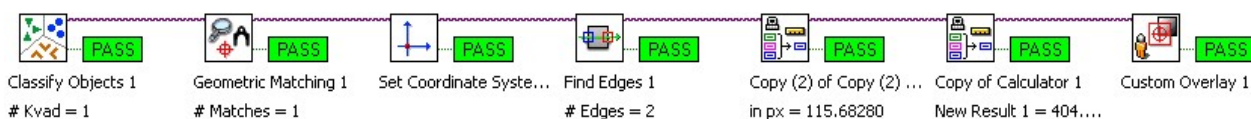


Рис.12 Функциональная последовательность распознавания квадрата на образце

При этом горизонтальные и вертикальные размеры и расстояния, полученные программой NI Vision Builder в пикселях, переводятся в миллиметры по формулам:

Формула, для перевода размеров объекта из пикселей в миллиметры:

$$L = (L_d * L * W) / (F * L_m)$$

где: W – размер объекта в пикселях.

L_д - размер датчика камеры в мм(4,376);

F – фокусное расстояние объектива (12 мм);

L_м - горизонтальное разрешение матрицы камеры в пикселях (640);

L – расстояние от матрицы камеры до объекта.

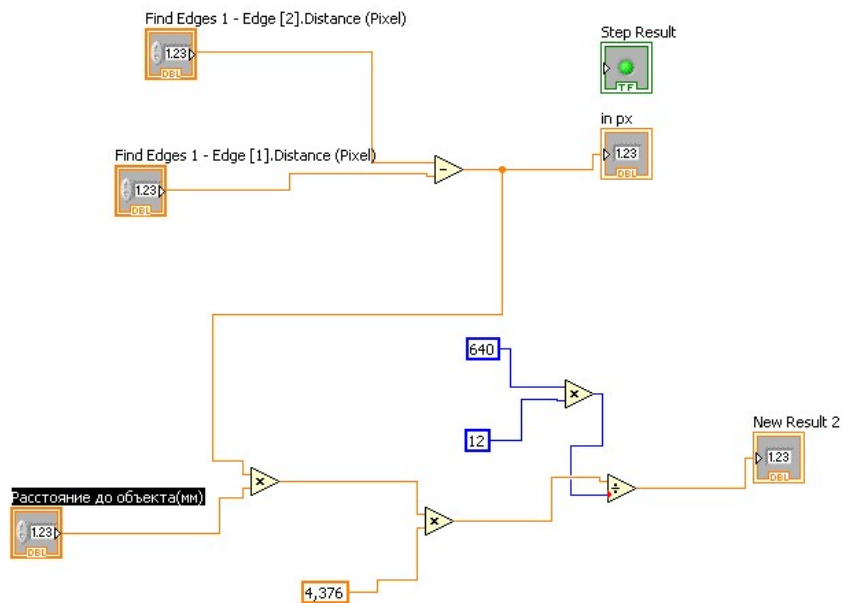


Рис.13 Перевод пикселей в мм

2.2 Если же объектов на изображении меньше или больше (но с корректно настроенной функцией, в рамках задания, больше их не может быть) программа переходит к последовательному поиску и определению принадлежности образца к одной из возможных фигур.

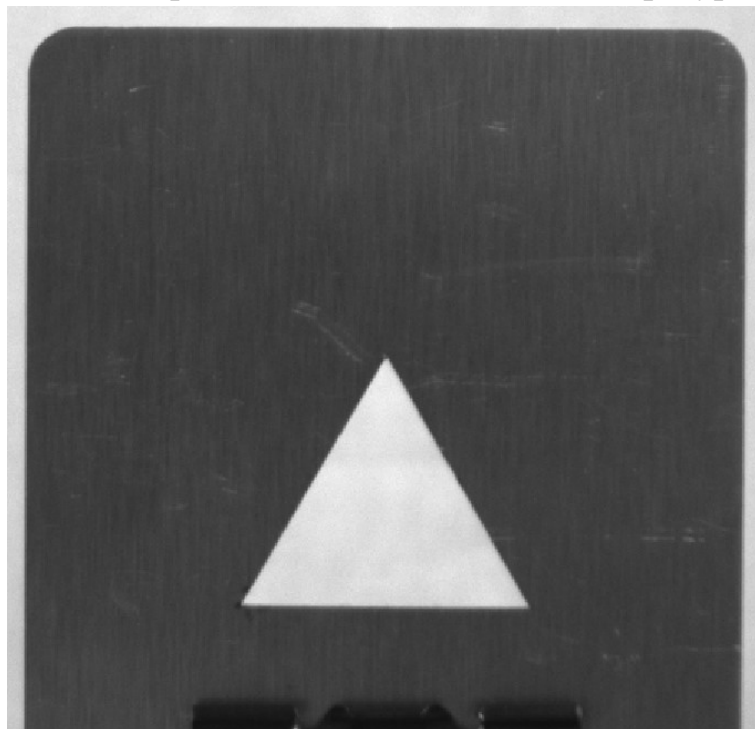


Рис.14 Получение изображения образца с одним объектом контроля

Дойдя до блока с распознаванием треугольника (в данном случае) объект будет опознан.

Затем в соответствии с опознанием образца будут сделаны необходимые измерения, в том числе и подсчет площади для любой из фигур в соответствии с формулами:

Треугольник(равносторонний): $S=a^3*1,71/4$

Где, а – грань треугольника

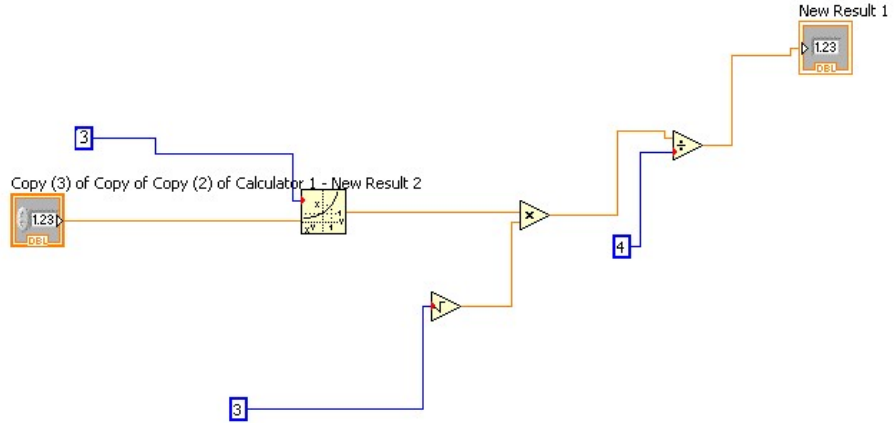


Рис.15 Вид формулы в калькуляторе

Квадрат: $S=a^2$

Где, а – грань квадрата

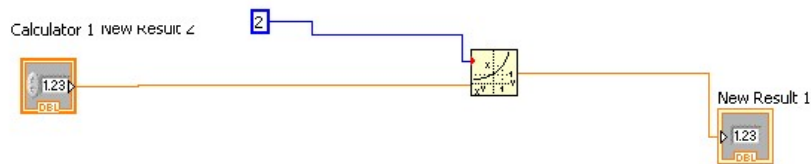


Рис.16 Вид формулы в калькуляторе

Круг: $S=2*\pi*R^2$

Где, R – радиус, π - математическая константа примерно =3.14,

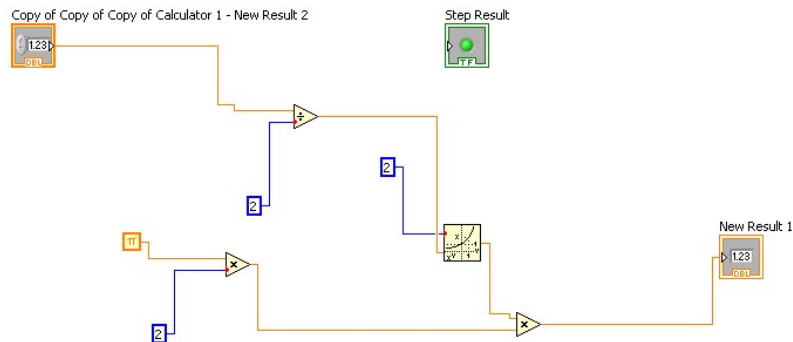


Рис.17 Вид формулы в калькуляторе

3. После всех вычислений программа выводит информацию о принадлежности образца к одному из видов (в случае с 3мя объектами информацию о нахождении всех 3х), а так же их параметры (линейные размеры и площади)

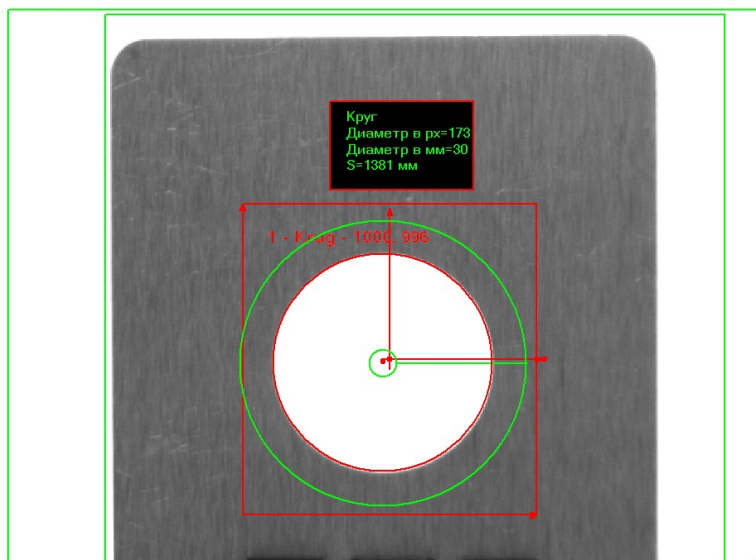


Рис.18 Вывод программой NI Vision Builder информации о найденном объекте.

Полученные с помощью программы данные позволяют облегчить задачи связанные с вычислениями и измерениями размеров и площадей объектов контроля: треугольник, квадрат, круг.

С помощью систем распознавания можно автоматизировать различные процессы, уменьшая при этом временные затраты и обеспечивая более высокий уровень надежности системы в целом. Самое важное в подобных системах это понятный и удобный интерфейс, который позволяет управлять процессами распознавания без какого либо знания языков программирования. Изучение подобных систем важно для более полноценного понимания того как происходят процессы распознавания в системах автоматического контроля.

Работа 1. Распознавание геометрических образов на базе камеры технического зрения NI 1722 в среде NI Vision Builder.

1.1. Цель лабораторной работы

Ознакомление с интегрированной средой NI Vision Builder

1.2. Теоретические положения.

NI Vision Builder for Automated Inspection представляет собой настраиваемое ПО для систем машинного зрения, предназначенное для моделирования, испытаний и внедрения приложений. Работа в NI Vision Builder for Automated Inspection осуществляется без программирования, однако позволяет реализовывать мощные приложения на уровне возможностей LabVIEW. Программа содержит встроенные средства сбора и анализа изображений и решения задач идентификации и сортировки. Кроме того, имеются возможности определения соответствия объекта шаблону, управления цифровыми линиями ввода/вывода и взаимодействия с внешними устройствами, например, промышленными контроллерами. Основные функции:

- Интерактивная оболочка разработки приложений машинного зрения
- Автоматическая генерация кодов LabVIEW
- Управление цифровым вводом/выводом, ПЛК и другими устройствами через последовательный порт
- Взаимодействие с устройствами ввода изображений компании NI для PCI, PXI и промышленными FireWire камерами.

Программный продукт NI Vision builder AI и интеллектуальная камера NI 17xx позволяют автоматизировать визуальный контроль на предприятии, исключив человека из процесса контроля, а также повысить скорость и точность операций контроля. NI Vision builder позволяет управлять интеллектуальной камерой, внешним источником света (вспышкой), а также различным промышленным оборудованием по протоколу Modbus, TCP/IP и serial ports.

Данная программная среда не требует освоения языка программирования и позволяет реализовать алгоритмы различной сложности с помощью блок-схем и несложных для понимания функций.

NI Vision builder AI позволяет получать и обрабатывать изображения непосредственно с интеллектуальной камеры и работать с изображениями, расположенными на локальном диске компьютера или сервера. Программа работает в двух режимах:

- режим конфигурации (configure inspection) позволяет создавать новые и изменять уже созданные алгоритмы обработки изображений. Этот режим позволяет использовать средства отладки алгоритмов.

- режим инспекции (inspect product) исполняет выбранный алгоритм и отображает результаты его работы.

1.2.1. Рабочее пространство конфигуратора

2. Панель запуска и отладки. В этой панели размещены кнопки:

- создание нового алгоритма.
- открытие существующего алгоритма.
- сохранение текущего алгоритма.
- кнопки масштабирования, доступные в том случае, если в рабочем поле просматривается изображение.
- запуск и остановка алгоритма.
- отладки алгоритма.
- группа кнопок для выбора изображения.

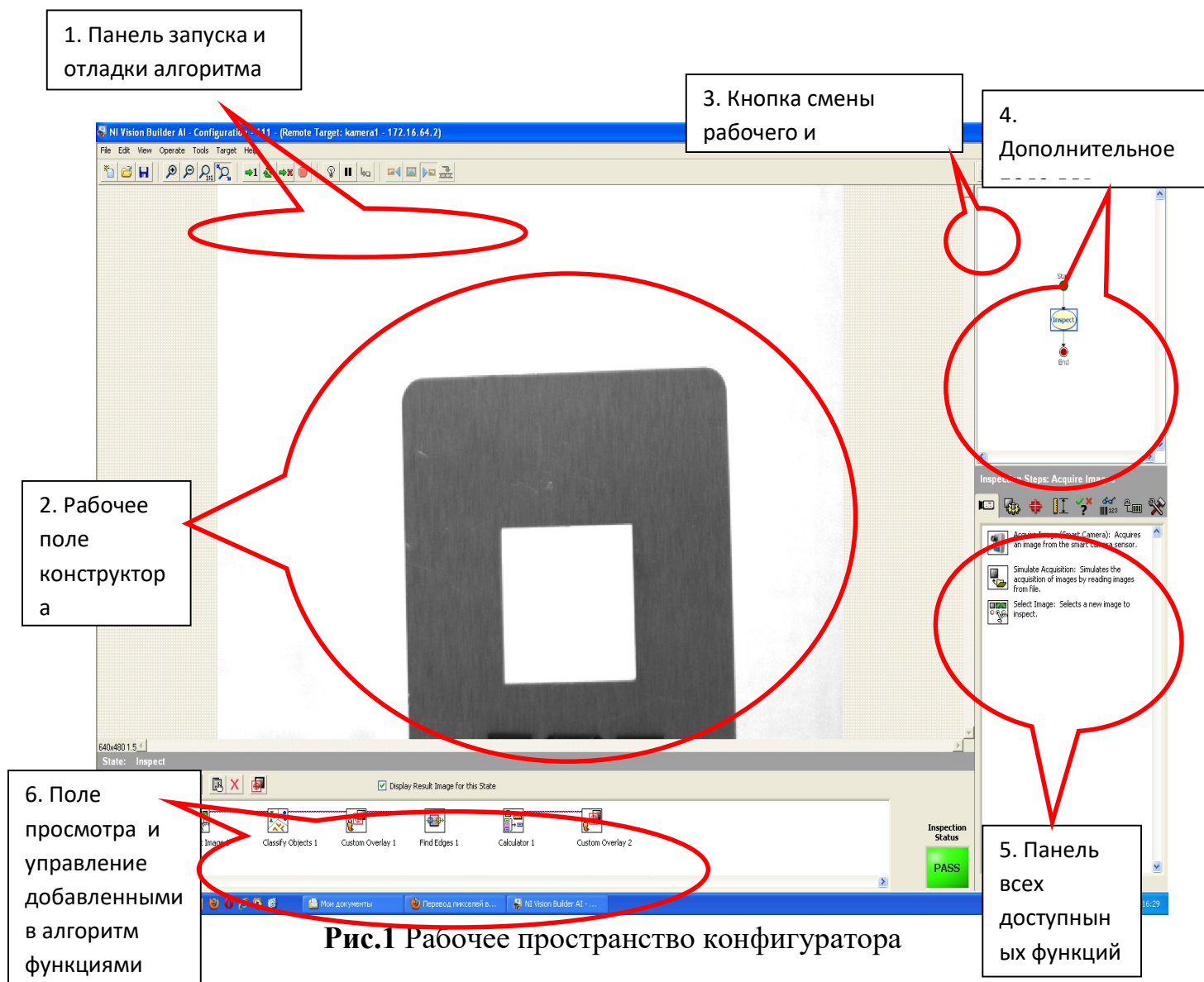


Рис.1 Рабочее пространство конфигуратора

3. Рабочее поле конструктора. В этом поле редактируется общий алгоритм(блок-схема). Все возможные действия доступны в контекстном меню. Кроме этого в этом поле изменяются параметры функций и отображаются результаты их выполнения.
4. Кнопка смены рабочего и дополнительного поля позволяет загрузить в рабочее поле общий алгоритм или обрабатываемое изображение.
5. Дополнительное поле для просмотра отображает уменьшенное обрабатываемое изображение или общий алгоритм(без возможности редактирования).
6. Панель всех доступных функций. В этой панели расположены все доступные функции, разделенные по категориям.
7. Поле просмотра и управление добавленными в алгоритм функциями. В этом поле отображаются функции для выбранного звена общего алгоритма(блок-схемы) и некоторые результаты их выполнения. Выбранную функцию можно редактировать, удалить, скопировать, вырезать, вставить и отключить через контекстное меню. Двойной щелчок по функции открывает в рабочем поле ее редактирование. Вид поля можно изменить в настройках: меню Tools>Options. В поле «Category» выбрать «Configuration», далее выбрать «Tableview» или «Diagramview».

Для работы с камерой в Vision builder Ai необходимо:

1. Подключить интеллектуальную камеру в локальную сеть, выбрать ее в списке доступных устройств, войти в режим конфигурации и выбрать функцию, которая позволяет просматривать текущее изображение с камеры.



Рис.2 Выбор функции получения изображения с камеры

2. Навести объектив камеры на интересующий нас объект и настроить фокус объектива, просматривая результат в рабочем поле Vision builder Ai. Для облегчения фокусировки можно положить рядом с объектом кусочек бумаги с текстом.
3. Отрегулировать яркость и контрастность изображения, после чего нажать кнопку «ОК». Функция добавится в поле 6. Для ее изменения

нужно щелкнуть по ней двойным щелчком левой кнопки мыши или выбрать пункт «Edit» в контекстном меню функции.

Если подключение камеры не требуется:

1. Необходимо загрузить изображение из файла:

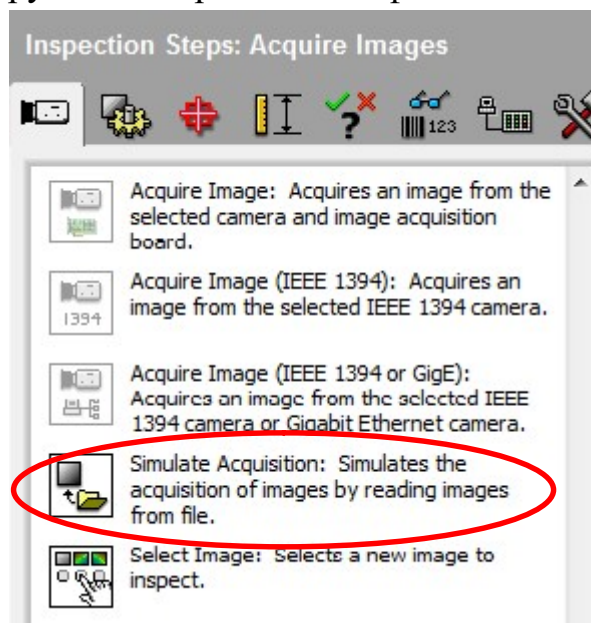


Рис. 3 Функция загрузки изображения

2. Выбрать файл изображения нажав на изображение папки:

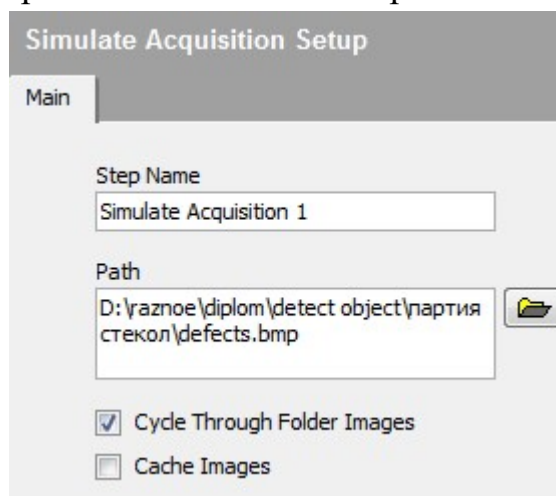


Рис. 4 Настройки функции

Если необходимо выбрать несколько изображений поочередно, то необходимо поставить галочку в пункте «CycleThroughFolderImages». При этом загружаемые поочередно изображения должны находиться в одной папке. Изображения будут меняться автоматически при каждом запуске алгоритма или можно изменять их вручную с помощью группы кнопок выбора изображения, расположенной в панели запуска и отладки алгоритма.

3. Нажать кнопку «ОК».

После этого необходимо выбрать изображение для дальнейшей его обработки:

1. Выбрать функцию

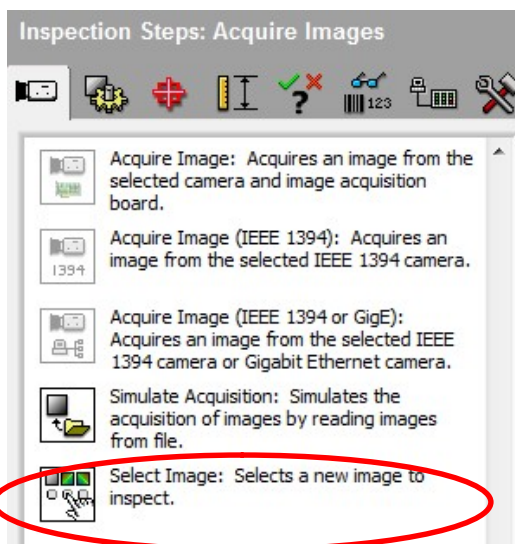


Рис.5 Функция выбора изображения

2. В открывшемся окне редактирования функции выбрать результат предыдущей функции и нажать «ОК»:

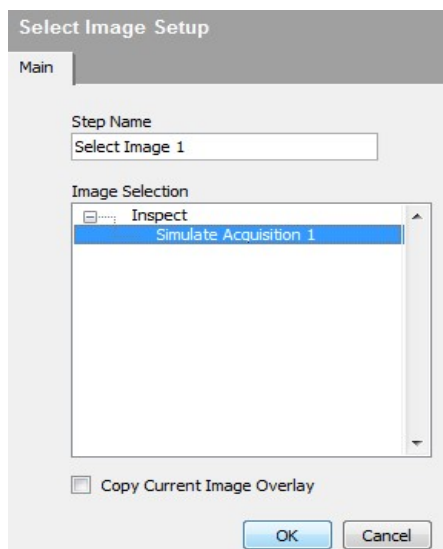


Рис.6 Настройки функции

Далее необходимо построить алгоритм решения задачи и воспользоваться необходимым для этого набором функций.

После того как алгоритм создан и отлажен его необходимо сохранить и закрыть конфигуратор Vision builder AI. После закрытия конфигуратора откроется окно выбора режима работы Vision builder AI:

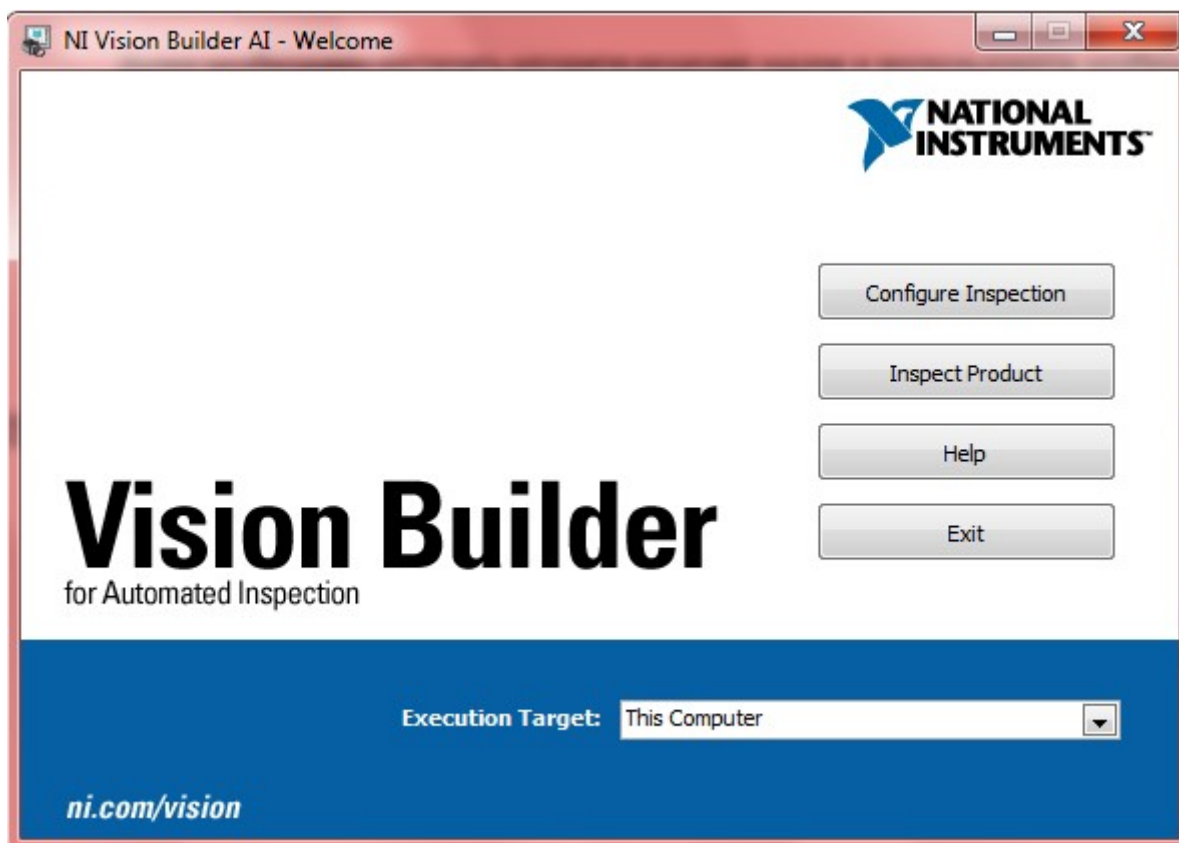


Рис.7 выбор режима работы Vision builder AI

В котором необходимо выбрать «InspectProduct», нажав соответствующую кнопку. Далее необходимо открыть сохраненный ранее файл инспекции, для этого в меню программы выбираем «Inspection>Openinspection». После этого запускаем открытый алгоритм: меню «Inspection>Startinspection». Остановить запущенный алгоритм можно воспользовавшись меню «Inspection>Stopinspection».

1.2.2 Создание общего алгоритма(блок-схемы)

Создание алгоритма происходит в рабочем поле конфигуратора. Если в рабочем поле находится изображение, то необходимо переключить содержимое рабочего поля кнопкой «Togglemainwindowview».

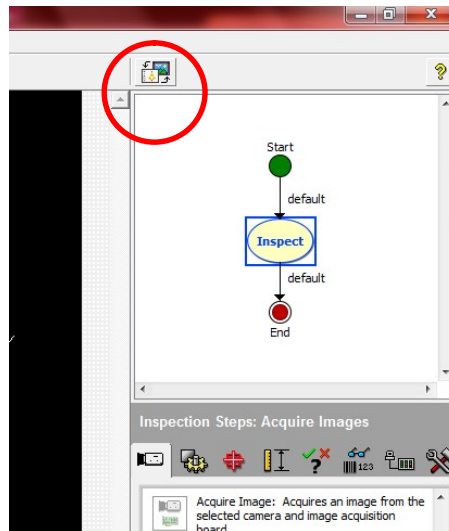


Рис.8 Кнопка смены поля

Поле того, как в рабочее поле загрузится схема простейшего алгоритма с единственным звеном «Inspect» можно приступить к созданию алгоритма.

Все действия с блоками («State») и переходами между блоков («Transition») доступны только из контекстного меню рабочего поля, блоков, связей и недоступны из главного меню программы.

3.2.3 Управление блоками («State»)

Алгоритм(блок-схему) сначала лучше нарисовать на бумаге, отметив в блоках примерные действия, которые в программе будут представлены функциями.

Для того, чтобы изменить имя блока необходимо вызвать на нем контекстное меню и выбрать пункт «rename» или щелкнуть на нем двойным щелчком правой кнопки мыши и ввести новое имя блока. После чего необходимо нажать «ОК» для сохранения нового имени.

- добавить новый блок можно вызвав контекстное меню в свободном месте рабочего поля и выбрать пункт «Createnewstate».

- удаление блока производится из контекстного меню, вызванного на нужном блоке и выбором пункта «Delete»

- для добавления и редактирования функций какого-либо блока необходимо выбрать его щелчком левой кнопки мыши. При этом в поле просмотра и управление добавленными в алгоритм функциями отобразятся все имеющиеся функции этого блока. Для добавления функции в блок ее необходимо выбрать из списка доступных функций, настроить и нажать «ОК».

1.2.4 Управление связями («Transition»)

Удалять и редактировать связи по-умолчанию(default) нельзя. Можно лишь изменять направление связи.

- для добавления новой связи необходимо вызвать контекстное меню на нужном блоке и выбрать пункт «CreateNewTransition», после чего указать второй блок для соединения.
- удалить связь можно вызвав на ней контекстное меню и выбрать соответствующий пункт «Deletetransition».
- для того, чтобы переместить связь на другой блок необходимо перетащить мышью конец связи на нужный блок.
- для редактирования связи необходимо дважды щелкнуть по ней правой кнопкой мыши или вызвать на связи контекстное меню и выбрать пункт «Edittransition». В открывшемся окне можно изменить имя связи и указать условие перехода по ней.

Пример: переход из «Блока 1» по «Связи 1» в «Блок 2» будет происходить только тогда, когда системное время компьютера будет превышать 19 часов. В противном случае переход будет происходить по связи «default» в конец блок-схемы («End») и функции «блока 2» выполняться не будут.

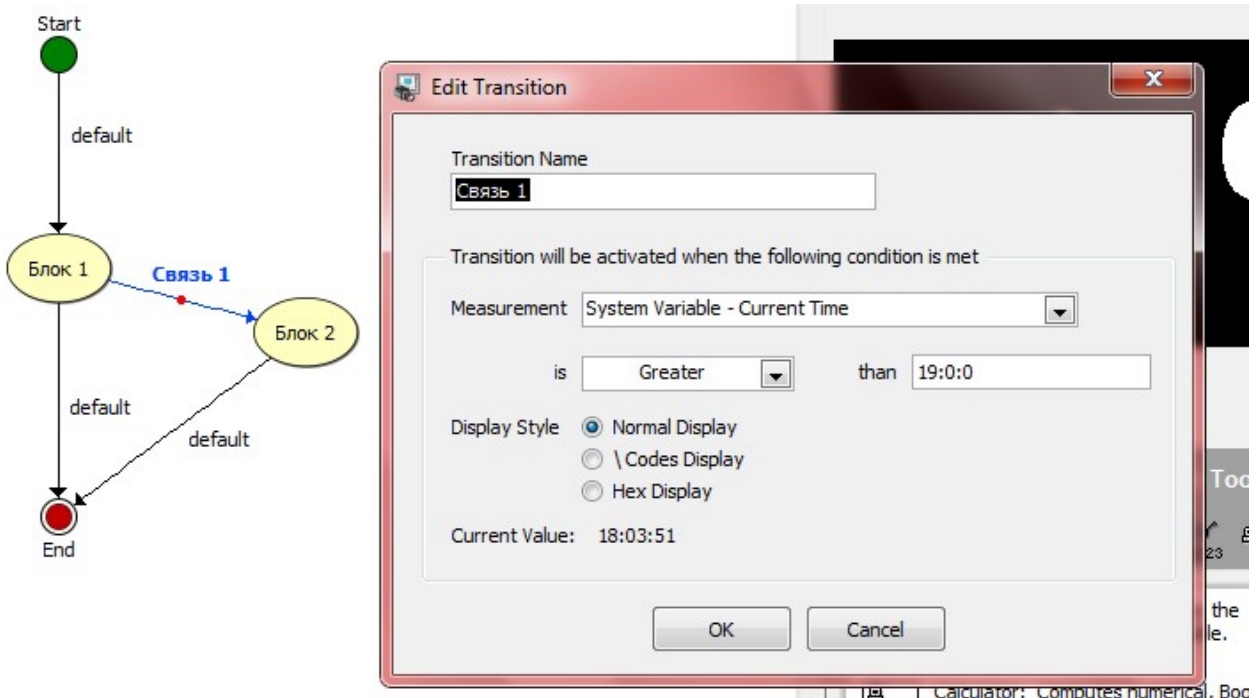


Рис.9 Управление связями

1.2.5 Создание и редактирование переменных

В Vision builder AI имеется возможность хранения промежуточных значений в переменных. Эти переменные видимы всем функциям на всех этапах работы алгоритма.

Переменные могут быть следующих типов:

- Numeric. Эти переменные могут хранить дробные или целые числа.
- Boolean. Значения этих переменных могут иметь лишь два значения «true» и «false» («истина» и «ложь»). Этот тип переменных используется в булевой алгебре.
- String. Этот тип предназначен для хранения строк.
- Point. Переменные этого типа предназначены для хранения двух координат и имеют две составляющие «X» и «Y».

Добавить новую переменную можно через менеджер переменных (Variable manager), доступный в главном меню «tools>variable manager». В открывшемся окне необходимо выбрать вкладку «Inspection variables» и нажать кнопку «Add». Далее необходимо ввести имя переменной (можно на русском языке), ее тип и начальное значение.

Изменить значение переменной можно в любом блоке алгоритма с помощью функции «Set Variable», расположенной на вкладке «Use Addition Tools»:

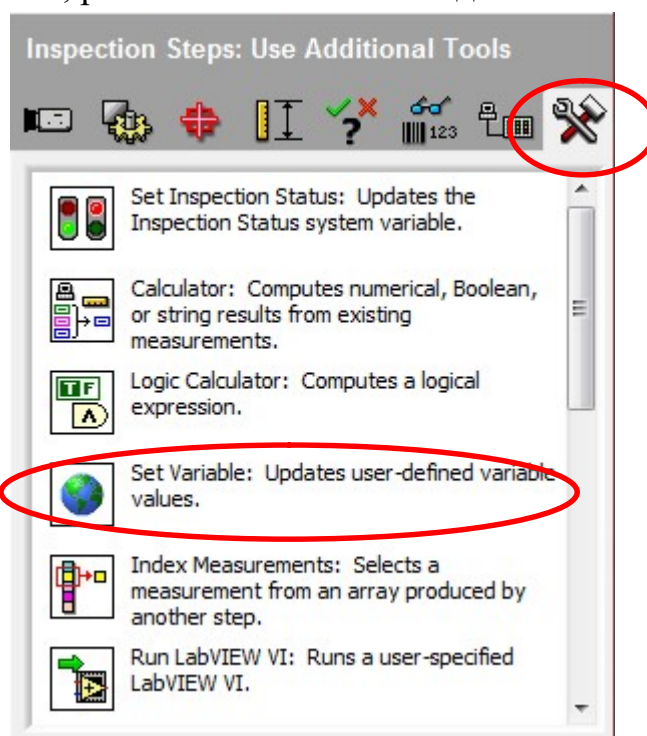


Рис.10 Выбор функции Set Variable

В окне редактирования функции «Set Variable» можно посмотреть текущие значения, а также установить новые значения переменных:

1. В таблице «Variables» отображаются имена переменных, их текущие значения, операции изменения и значения после изменения.
2. Ниже таблицы располагается панель «Operation», на которой можно установить новые значения переменных следующими способами:
 - «DonotSet» - переменная не будет изменена.
 - «SettoConstant» - установить явное значение переменной.

- «SettoMeasurement» - установить значение переменной как выходное значение какой-либо предыдущей функции или приравнять ее другой переменной.
- «Increment» - увеличить значение переменной на единицу (доступно только для переменных типа «Numeric» и «Point»).
- «Decrement» - уменьшить значение переменной на единицу (доступно только для переменных типа «Numeric» и «Point»).

3. Кнопка «Edit Variables» открывает менеджер переменных (Variable manager).

После настройки функции необходимо нажать кнопку «ОК» для сохранения изменений или «Cancel» для отмены.

Функция «калькулятор»

Данная функция предназначена для различных арифметических и логических операций, а также операций со строками. В качестве входных значений могут быть выбраны переменные или результаты выполнения предыдущих функций. Результатом работы этой функции может быть ряд выходных значений, которые могут быть переданы в переменные с помощью функции «Set Variable» или использоваться в последующих функциях.

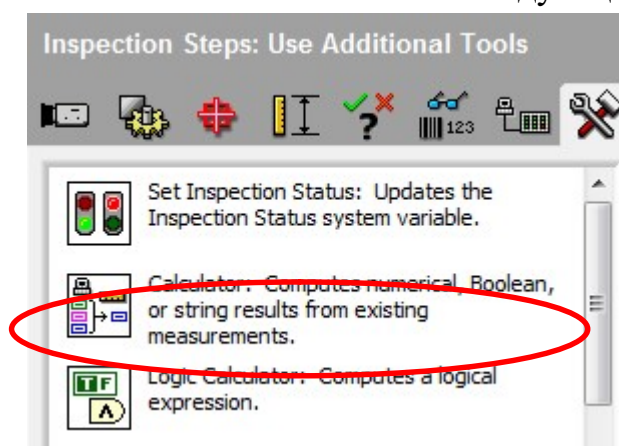


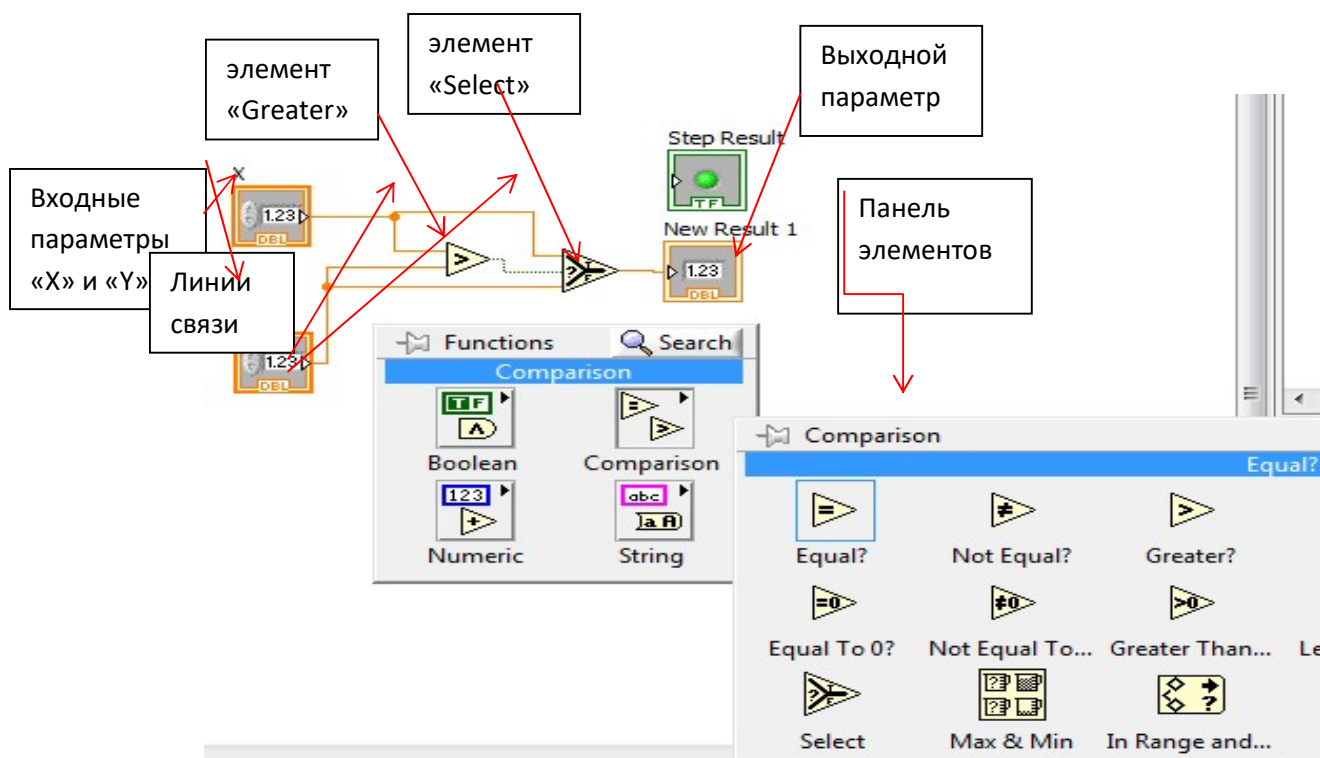
Рис. 11 Выбор функции Calculator

После выбора функции запустится мастер настройки, с помощью которого можно выбрать входные и выходные значения функции. Далее можно приступить к созданию цепочки вычислений. Цепочка вычислений представляет собой совокупность различных элементов, выполняющих определенные действия, и линий связи между этими элементами.

Пример: выбор наибольшего из двух чисел.

В качестве входных параметров выбраны две переменные «X» и «Y», которые сравниваются между собой при помощи элемента «Greater». На выходе этого элемента появляется сигнал «true» (истина), если «X» больше «Y» или «false» (ложь), если «X» меньше либо равно «Y». Далее по схеме

следует элемент «Select», на выходе которого будет наибольшая по величине переменная. Значение большей переменной заносится в выходной параметр «NewResult 1» функции калькулятор.



Функция «классификации объекта»



Classify Objects: Classifies particles in a region of interest based on their shape.

Функция классификации объектов, предназначена для определения нужного объекта из множества отображенных на снимке с камеры. Она имеет возможность отсеивать мелкие объекты, что несомненно помогает при определении более крупных по размерам фигур. Поле настройки данной функции выглядит можно увидеть на рис. 4:

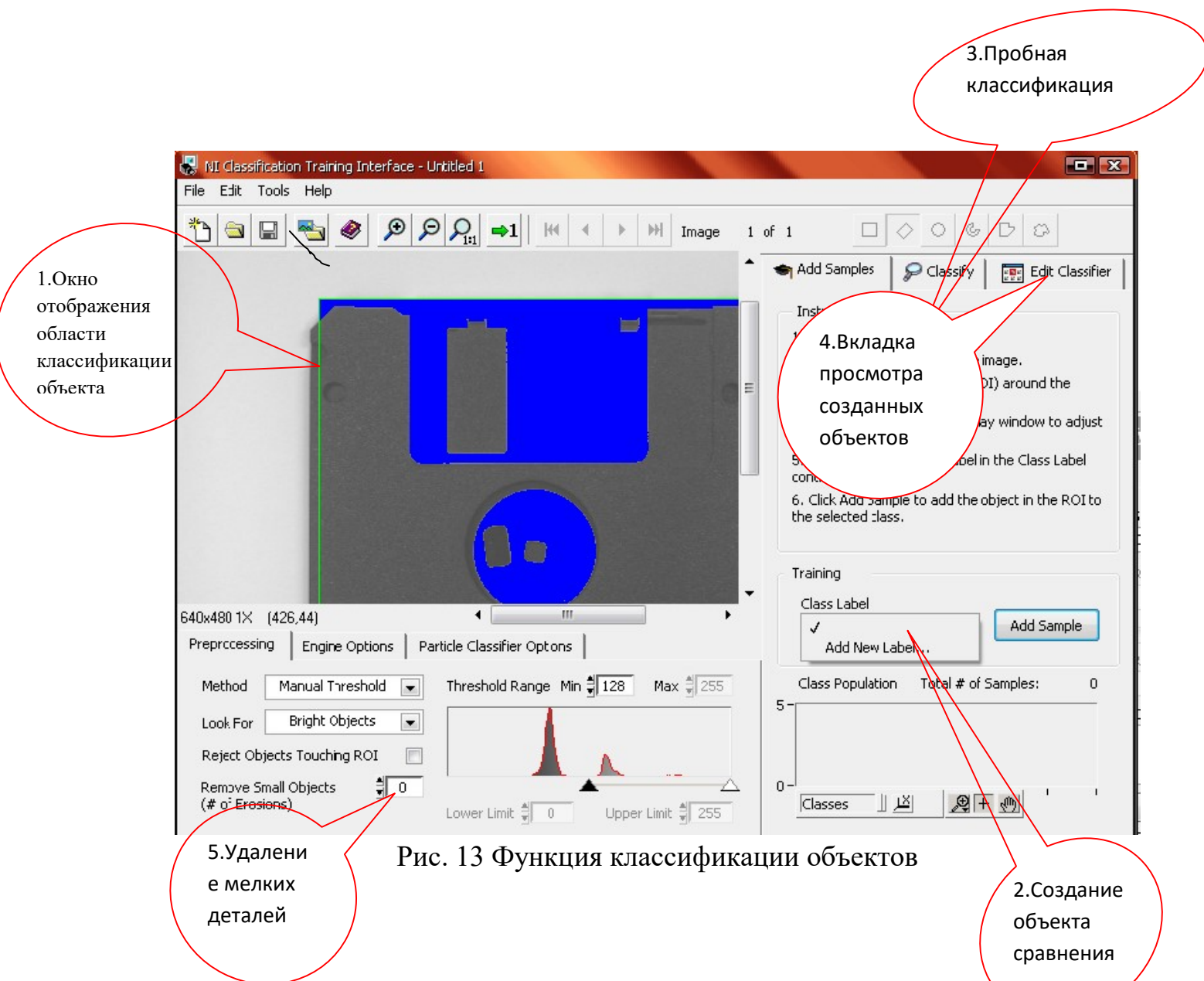


Рис. 13 Функция классификации объектов

1.- В этом окне отображается та зона в которой вы указали искать объект распознавания, на нем видны все объекты которые были найдены, для настройки поиска нужных объектов можно использовать различные методы (Method)-находящиеся под окном распознавания, так же можно уточнять светлые или темные объекты вы ищите Look for-расположена рядом с Method.

2.-Создает объект сравнения, к нему функция будет обращаться для того что бы в будущем сравнивать с изображением с камеры, его нужно создать когда вы уже настроили окно распознавание на нужный вам объект, по сути эта создание эталонной функции и присвоения ей имени.

3.-Эта вкладка нужна для того что бы посмотреть находит ли функция тот объект что вы создали в пункте 2.

4.-В этой вкладке видны все созданные вами объекты сравнения и их Sample's, так называются эталоны.

5.-Зачастую в область распознавания попадают различные мелкие дефекты, этот пункт вам поможет настроить окно отображения так что бы функция видела нужный вам объект.

Функция «нахождения линейного размера»



Find Edges: Locates and counts intensity transitions along a line in the image.

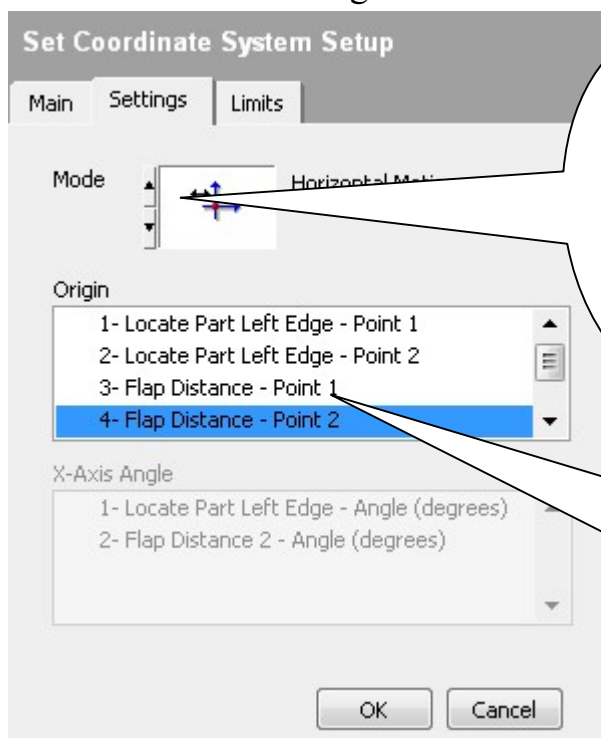
-Функция нахождения линейного размера, именно с ее помощью мы и определяем нужные нам величины, правда она их выдает лишь в пикселях, после чего функция калькулятор помогает нам придать значению более привычный и понятный вид. Для работы с изображением которой берем с камеры желательно привязывать размеры к осям координат, для этого функцию Set Coordinate System следует ставить да размера. Тогда при создании функции размера поле Reposition Region станет активным и нужно будет поставить в нем галочку.

Функция «выбора системы координат»



Set Coordinate System: Builds a coordinate system based on the location and orientation of a reference feature.

-Одна из самых важных, на мой взгляд, функций представленных здесь, выбор системы координат, она необходима, что бы привязать измерения величин объекта к самому объекту, иначе нам пришлось бы все время четко в одно, и тоже место ставить объект контроля. Для того что бы координатная система изменялась вслед за изменением положения объекта необходимо воспользоваться функцией Geometric Matching.



Поле для выбора смещения координатных осей

Поле выбора точки к которой будет привязана координатная система

Рис.14 Настройки функции

Функция «геометрического соответствия»



Geometric Matching: Locates grayscale features based on edge information in the entire image or a region of interest.

-Данная функция предназначена для нахождения в интересующей вас области объекта, к которому и будет привязана эта функция.

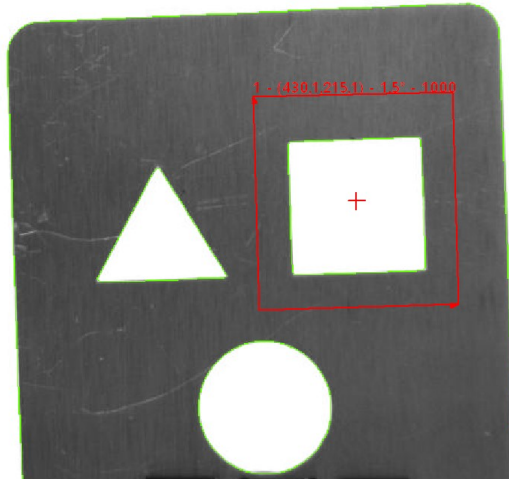


Рис.15 Окно определяемого объекта

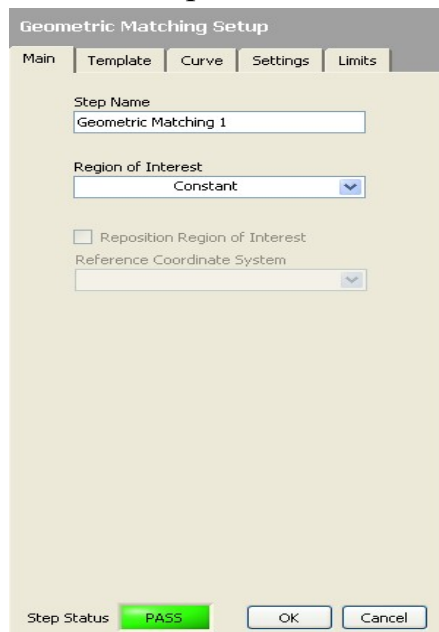


Рис.16 Окно параметров данной функции.

Функция «логический калькулятор»



Logic Calculator: Computes a logical expression.

-Функция логического калькулятора предназначена для выявления истинности, каких либо циклов что уже произошли, так же с ее помощью можно реализовать переход от одного блока алгоритма к другому.

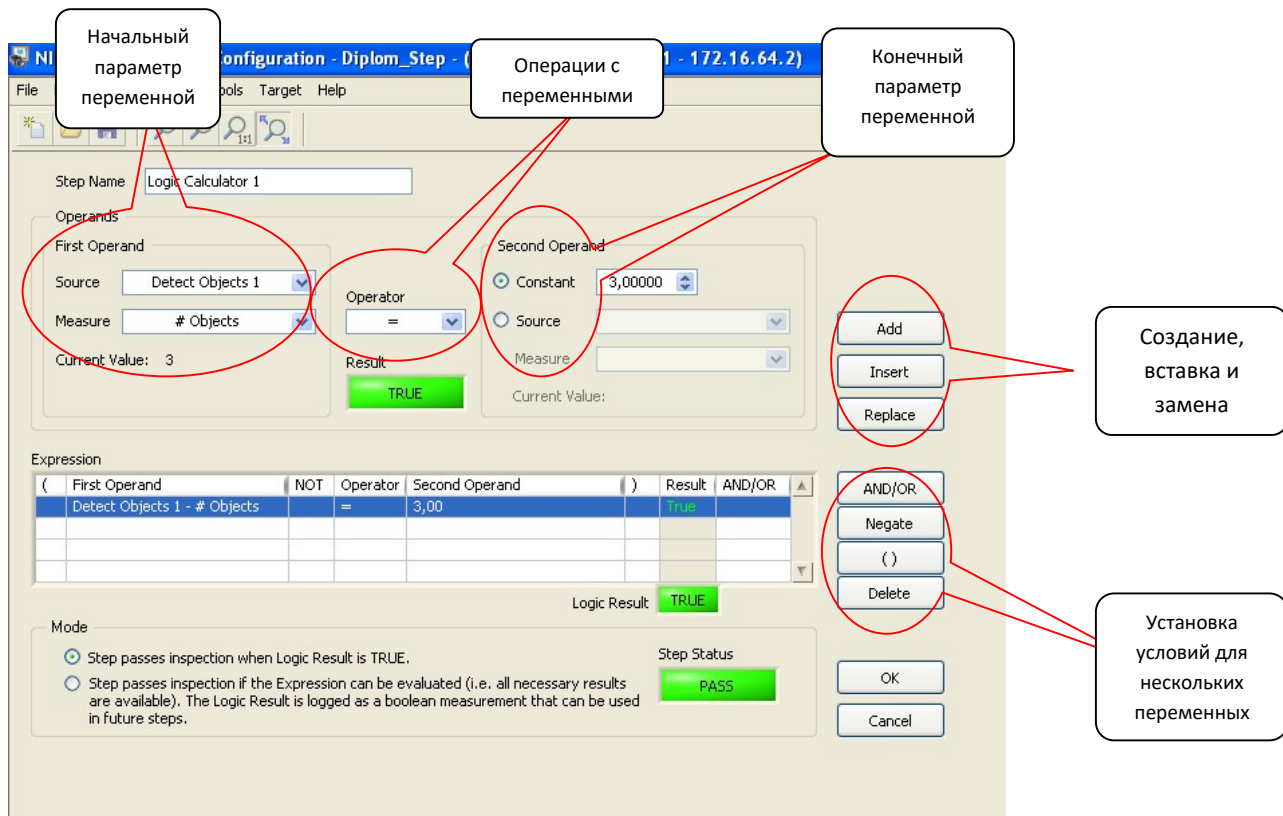


Рис.17 Окно настройки логического калькулятора

Функция «поиска размеров окружности»



Find Circular Edge: Locates a circular edge in a region of interest.

-Функция нахождения параметров окружности, в ней можно задавать диапазон поиска окружностей от минимального радиуса до максимального, очень удобная функция для поиска отверстий круглой формы.

Функция «отображения результатов»



Custom Overlay: Overlays custom data onto the current image.

-Очень удобная функция, предназначенная для отображения полученных результатов в предыдущих операциях. С ее помощью выводятся на экран размеры и наименование распознанного объекта:

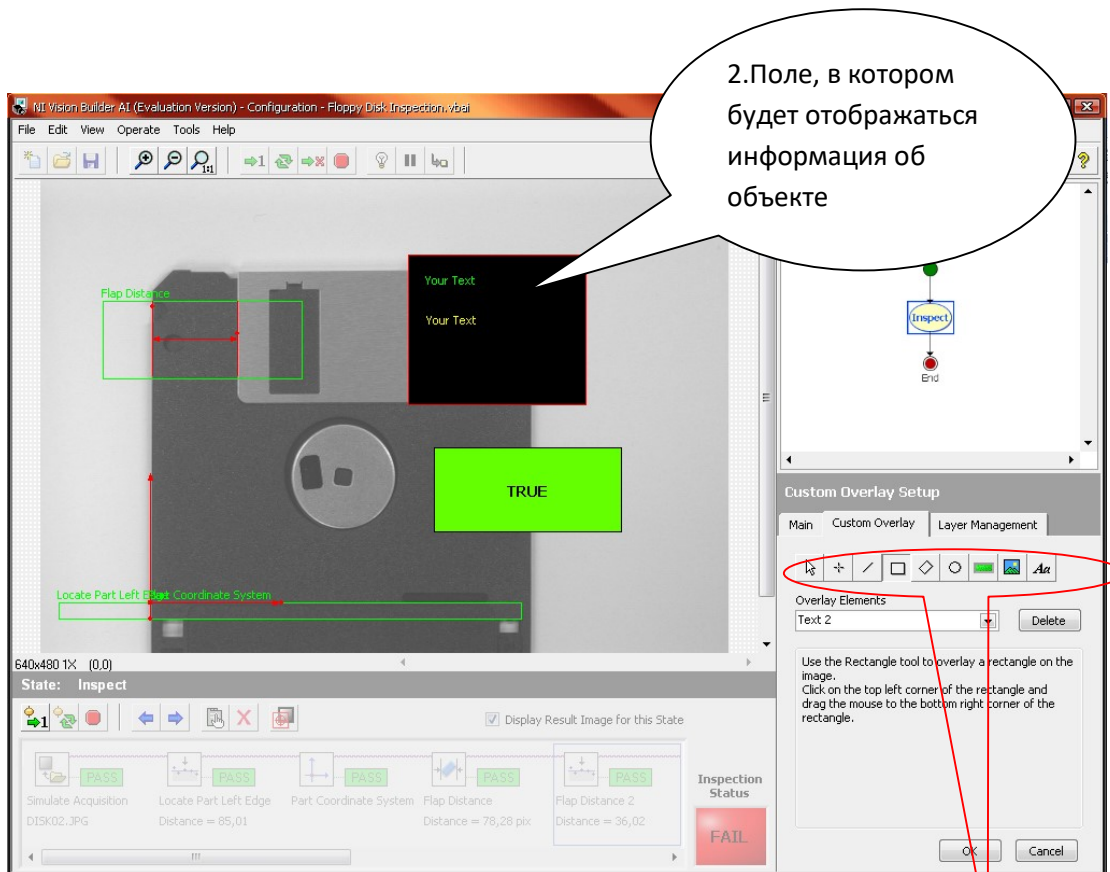


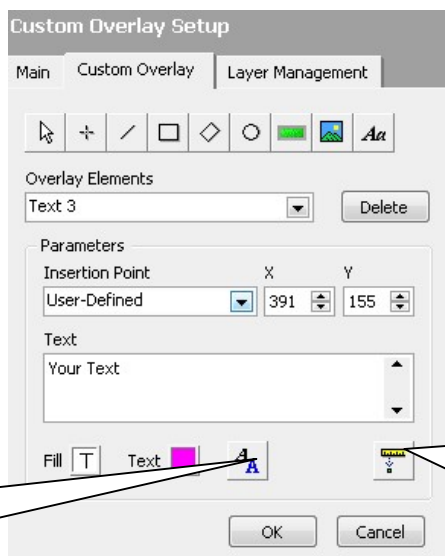
Рис.18 Отображение результатов

1. Инструменты для создания полей и текстов отображения

1.-В этом поле находятся средства для перемещения, как текста, так и поля отображения, так же создания самих полей и текстов.

2.-Поле отображение информации можно настраивать, менять его размеры, цвет границы и внутренней заливки.

После выбора места, где будет отображаться текст, нужно настроить сам отображаемый текст:



1. Параметры текста

2. Привязка текста к предыдущим функциям

Рис.19 Окно настройки отображаемого текста

- 1.-В этом меню мы можем настраивать расположение текста, стиль, эффекты, а так же вид. Причем для отображения русскоязычной речи в поле Desired Font нужно выбрать параметр Dialog Font.
- 2.- Здесь можно выбрать какие параметры и каких именно функций будут выводиться на экран.

Работа №2. Определение размеров объектов контроля

Для начала работы подготовим рабочее место. Перед запуском NI Vision builder AI v. 3.6 следует собрать лабораторный стенд, причем крепление камеры нужно расположить в 350 мм от крепления объекта. Закрепив объект с камерой на оптической линейке, следует подключить камеру непосредственно к компьютеру, не забыв до этого включить ее в сеть. Перед тем как начать работать рекомендуется прочесть пункт «Общие сведения»

2.1. Выполнение работы

Когда все приготовления сделаны можно начать работать с NI Vision builder AI v. 3.6. Для начала напишем программу для распознавания треугольника. Следовательно, в качестве объекта берем пластину с треугольником и закрепляем ее на стенде. Теперь можно приступить собственно к написанию программы, в рабочем пространстве конфигуратора переходим к окну с алгоритмом, нажав кнопку «смены рабочего и дополнительного поля». В этом окне нужно создать дополнительный блок, как показано ниже:

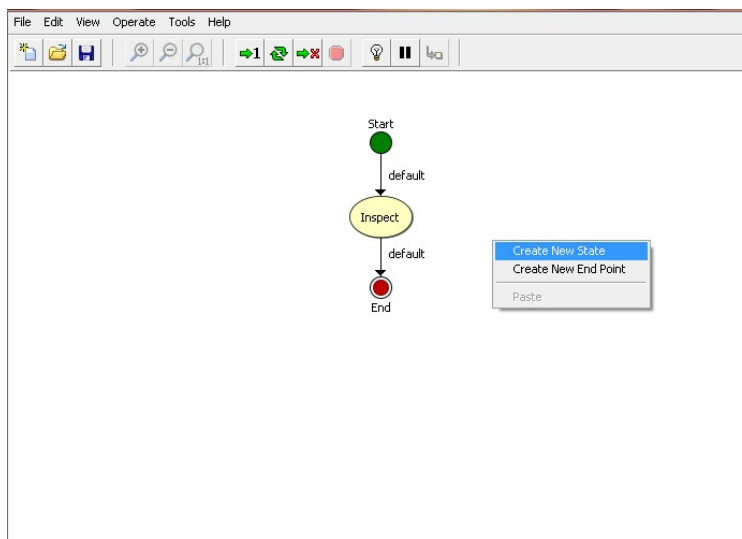


Рис. 1 Окно построение функциональных блоков алгоритма

После чего вам предложат назвать новый блок:

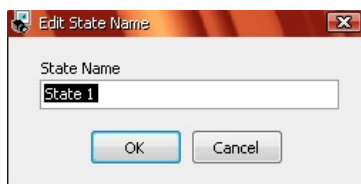


Рис.2 Название блока

Назовите его к примеру: «Распознавание треугольника».

Теперь переименуйте другой блок и установите связи между блоками, что бы получить один алгоритм:

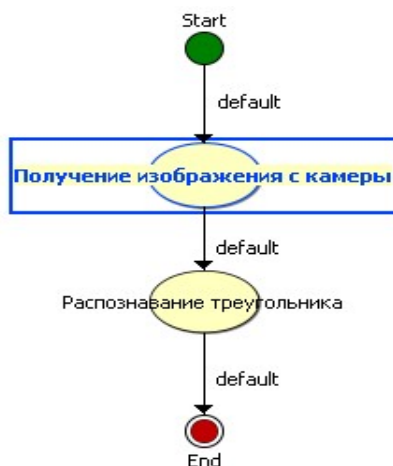
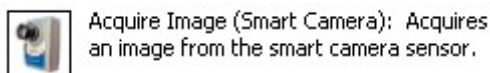


Рис.3 Вид получившегося алгоритма

У вас есть небольшой алгоритм, но его блоки пока не имеют никаких функций, теперь перейдем к созданию самих функций. Для этого воспользовавшись кнопкой «смены рабочего и дополнительного поля» перейдем на рабочее поле конфигуратора. На панели доступных функций выберем вкладку «Acquire Images». Щелчком по дополнительному полю нужно выбрать блок «Получение изображения с камеры», затем в открывшейся вкладке выбрать функцию-



С её помощью мы сможем получать изображение с камеры в любой момент, но перед тем как это сделать саму камеру необходимо сфокусировать на объекте, для этого в данной функции есть возможность просмотра изображения в реальном времени. Нажав, запуск вы увидите получаемую картинку и сможете настроить фокус камеры, так как это нужно.

По завершению настройки фокуса нужно создать данную функцию нажав «Ок». После чего вы увидите что в «Поле просмотра и управление добавленными в алгоритм функциями» появилась наша функция.

После чего можно переходить ко второму блоку алгоритма. Так как изображение мы уже получили, то можем использовать функцию, выбрав ее в рабочем поле появится зеленого цвета контур, который можно перемещать

и увеличивать, выделите рамкой всю пластину, после чего вам нужно будет создать новый файл классификации:

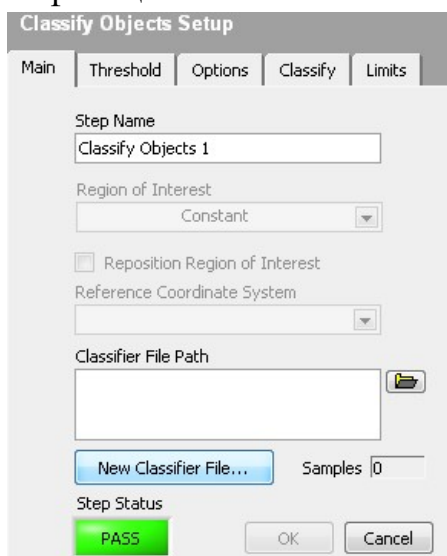


Рис.4 Панель настройки блока

Для этого нажмем «New Classifier File», после чего появится окно с настройками классификации объекта. «Classify Objects» см.рис. 4, в поле Class label создайте новую метку, предварительно настроив область классификации так что бы определялась только наша фигура и назовите ее «Треугольник»(русский шрифт не отображается). Затем перейдите во вкладку Classify:

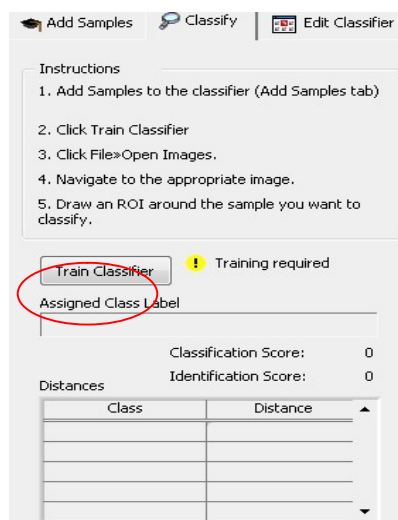


Рис. 5 Вкладка пробной классификации образца

Нажмите на отмеченную кнопку и, если в столбце Distance появится 0 то все хорошо. Вкладка Edit Classifier предназначена для хранения созданных нами образцов классификации, так же их можно удалять.

Раз уж мы определили к какому классу относится наш объект стоит перейти к измерению его величин. Для начала создадим функцию геометрического соответствия (см. пункт функция «геометрического соответствия»). Отметим наш треугольник в качестве объекта

геометрического соответствия (ГС), на экране рабочего поля появится красный контур. Далее привяжем функцию выбора системы координат к полученному объекту ГС (см. пункт функция «геометрического соответствия»)

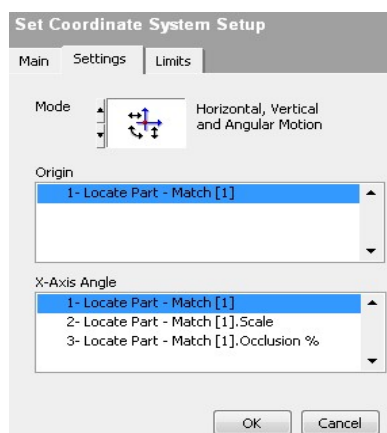


Рис.6 Настройка координатных осей

Выбрать параметры как указано выше.

После привязки к объекту координатных осей, можно приступить к измерениям, в случае с треугольником мы будем измерять длину его стороны и высоту треугольника. На панели функций открываем вкладку Locate Features и выбираем функцию Find Edges(она понадобится дважды), для начала строим прямую от вершины треугольника к основанию, после этого заново выбираем эту функцию и проводим прямую вдоль одной из сторон, так что бы появилось две точки.

После нанесения и привязки размеров к объекту, нам понадобится функция калькулятор, ввиду того что мы проделали 2 измерения придется дважды переводить пиксели в миллиметры, что бы получить обе величины в привычном для нас виде. (см. пункт Функция «калькулятор»)

Выбрав ее мы перейдем в окно редактора этой функции, где необходимо создать цепочку вычислений согласно заданию, в нашем случае это перевод пикселей в миллиметры:

$$L = (L_d * L * W) / (F * L_m)$$

где: W – размер объекта в пикселях.

L_д - размер датчика камеры в мм;

F – фокусное расстояние объектива (12 мм);

L_м - горизонтальное разрешение матрицы камеры в пикселях (640);

L – расстояние от матрицы камеры до объекта.

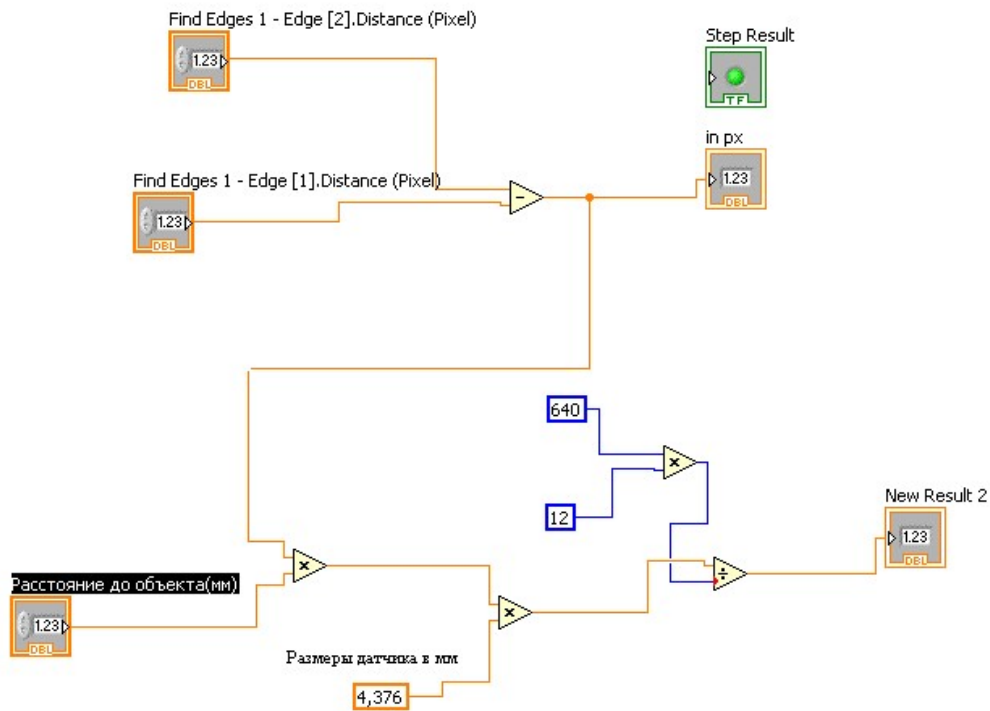


Рис.7 Реализация формулы в калькуляторе

Так выглядит стандартная формула перенесенная на язык понятный нашей функции, подобным способом можно реализовать различные вычисления. Входные значения получаем путем нажатия кнопки Edit Inputs\Outputs, далее ставим галочки в блоке Find Edges 1(высота) на против 2х значений Distance(соответственно это будут координаты 1ой точки и 2ой), причем для нахождения корректной величины нужно из 2го вычитать 1ое, иначе получите отрицательный результат.

Входное значение «Расстояние до объекта» измеряются заранее самостоятельно при установки камеры и объекта на линейку. Создаём переменную, вбиваем в нее свою величину(рекомендуется равную 305). Затем щелкаем правой кнопкой мыши по ней и в пункте Representation выбираем DBL, и меняем нашу переменную на Control. Теперь создаем еще один калькулятор с точно такой же функцией за одним лишь исключением, входами выбираем Distance при блоке Find Edges 2(сторона треугольника). Теперь найдя два линейных размера мы можем измерить к примеру площадь треугольника, так как у нас он равносторонний, для вычисления будем пользоваться формулой площади для равностороннего треугольника:

$$S = a^3 * 1.71 / 4$$

Где, а - длинна грани треугольника

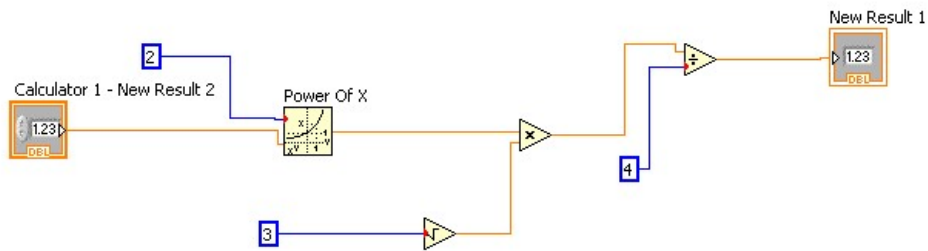


Рис.8 Реализация формулы в калькуляторе

Блок Power of X является аналогом возведение числа в степень, X возводится в степень Y, X – нижний вход блока.

Сделав все нужные вычисления можно перейти к выводу информации на экран. Для этого используем функцию Custom Overlay:

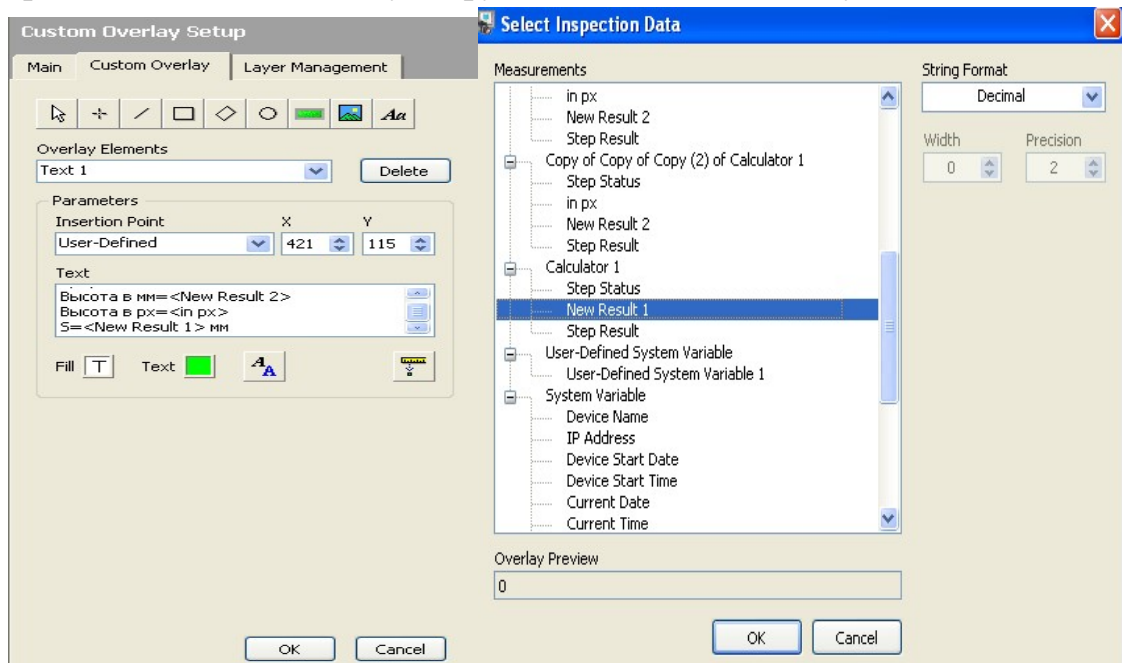


Рис.9 Настройки блока Custom Overlay

После того как выберем место где будет выводиться информация об объекте и настройка всей выводимой информации, можно опробовать получившуюся программу поменяв положения нашего треугольника, нажав на рабочем поле, кнопки Run. Затем сохраните написанную вами программу.

Работа №3. Классификация объектов

3.1 Выполнению задания

Нам предстоит дополнить уже имеющуюся первую лабораторную работу, добавив в нее распознавание следующего образца – квадрата. Для этого нам нужно вновь перейти на поле с функциональными блоками, переименуем блок «Распознавание треугольника» в «Измерение параметров треугольника», затем создадим еще 3 блока: «распознавание треугольника», «распознавание квадрата» и «измерение параметров квадрата», для создание новых связей между блоками, щелкните по блоку правой кнопкой мыши, в итоге алгоритм должен принять вид:

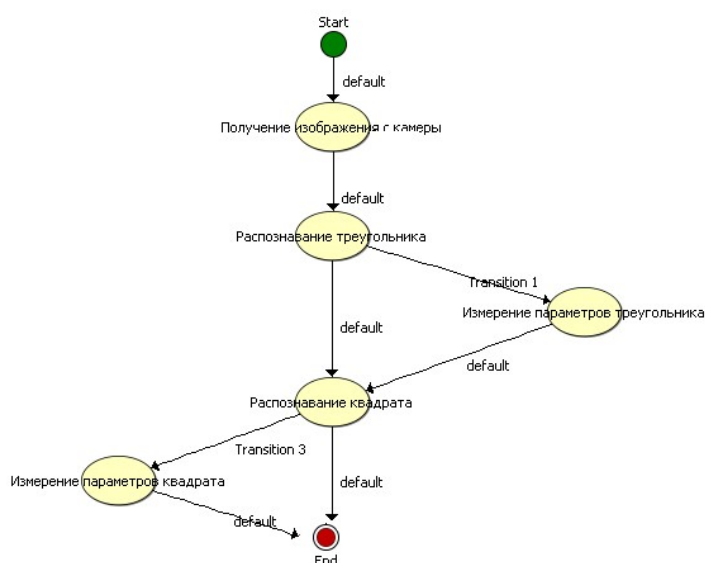


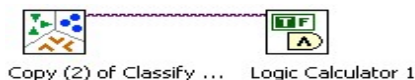
Рис.1 Алгоритм управляющей программы

Как говорилось выше, мы переименовали один имеющийся блок и получили в блоке с названием «Измерение параметров треугольника» нашу первую работу:



Рис.2 Вид функционально последовательности в блоке

Теперь переходим к блоку «Распознавание треугольника», в нем нужно будет создать следующие функции:



Первую функцию нужно настроить так же как в первой работе на распознавание треугольника, второй же блок у нас логический калькулятор, с его помощью мы сможем реализовать переход к нужным нам блокам алгоритма. (см. пункт Функция «логический калькулятор») В калькуляторе установить значение треугольника равным 1 и добавить в строку созданных условий. Затем возвращаемся на дополнительное поле с алгоритмом, на созданную нами связь между блоками «распознавания треугольника» и «измерением параметров» щелкаем правой кнопкой мыши и выбираем Edit Transition и там устанавливаем:

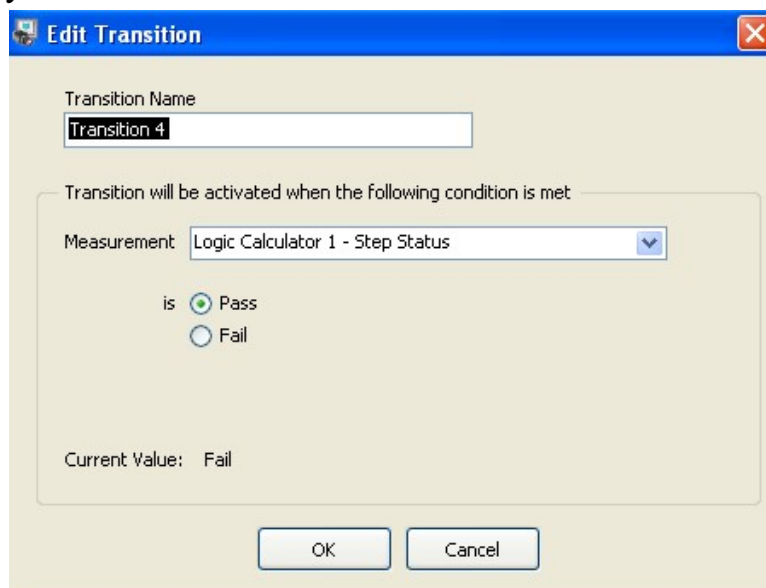


Рис.3 Правка связи

После чего можете проверить распознавание треугольника, определив в блоке что образец – треугольник программа пойдет в блок с права и только потом к «распознаванию квадрата».

В блоке «распознаванию квадрата» проделываем то же, что и для треугольника, только предварительно устанавливаем в зажим образец с квадратом и настраиваем блок распознавания объекта относительно новой фигуры, для квадрата рекомендуется установить удаление мелких объектов на 30. Если возникнут проблемы, и фигура будет распознавать некорректно (путать треугольник с квадратом), следует более точно подбирать параметры распознавания так же можно добавить несколько Sample's(кадров) меняете положение объекта, получаете снимок и в настройке распознавания нажимаете Add Sample.

Далее переходим к созданию функций в блоке «измерение параметров квадрата», функции используются аналогичные первой работе:

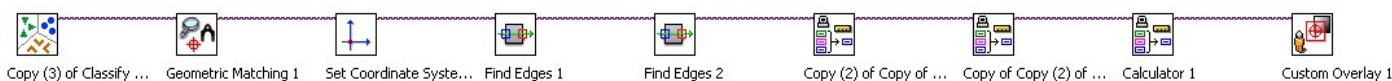


Рис.4 Вид функциональной последовательности блока

Настраиваем блок Classify Object так же как и в предыдущем блоке для квадрата, далее определяем расположение квадрата и привязываем оси к блоку Geometric Matching. Далее переходим к установке размеров, вначале мерим диагональ квадрата, затем длину его стороны, первые два калькулятора это перевод пикселей в миллиметры, а третий у нас будет считать площадь квадрата по формуле $S=a^2$:

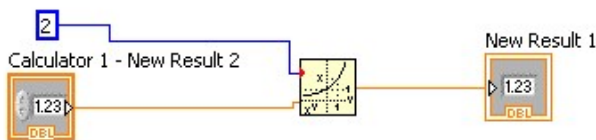
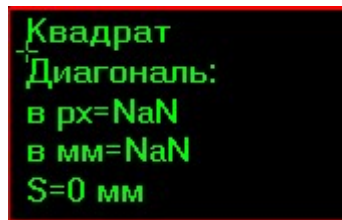
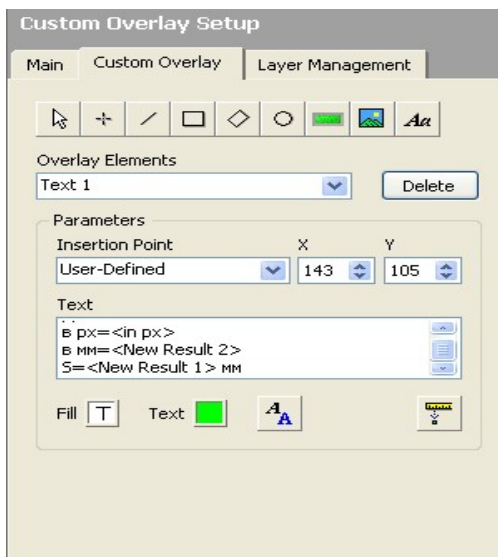


Рис.5 Нахождение площади квадрата

И в конце нашей программы реализуем вывод информации о квадрате, как было сделано в лабораторной работе №1 по средствам функции Custom Overlay.

-Визуальный пример отображения рамки с текстом



-Настройки параметров текста для квадрата (см. пункт Функция «отображения результатов»)

Написав программу, можете проверить ее работоспособность, не забудьте сохранить результат.

3.5 Лабораторная работа №3

Задание: напишите управляющую программу для распознавания образцов контроля(треугольник, квадрат, круг) и определения их параметров: треугольник – определить высоту и площадь, квадрат - определить диагональ и площадь, круг- определить диаметр и. Прделав предыдущие работы, вам останется лишь добавить с систему распознавания элементы определения окружности. Алгоритм примет вид:

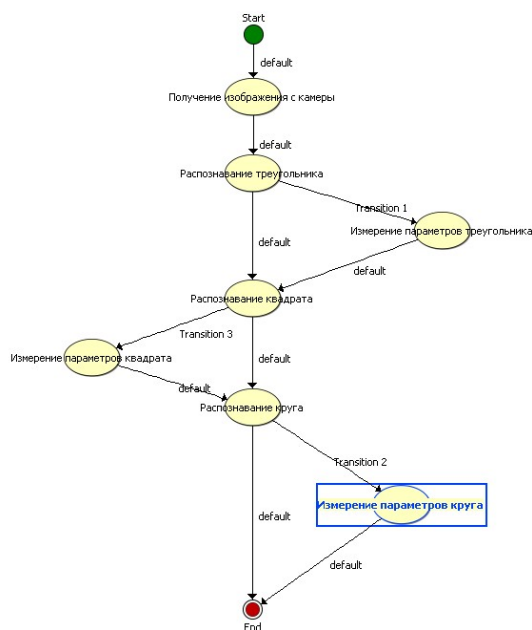


Рис.1 Алгоритм в среде NI Vision builder AI v. 3.6

Блок «Распознавание круга» настраивается аналогично квадрату и треугольнику. В блоке «измерение параметров круга» так же делаем все подобно предыдущим работам, однако для определения диаметра окружности пользуемся не линейными измерениями, а функцией Find Circular Edges, предназначенную специально для работы с окружностями.

Так же нам понадобится калькулятор, но в отличие от предыдущих вариантов, на один меньше, так как нам не нужно переводить несколько величин в миллиметры, хватит перевести только радиус окружности, а дальше использовать формулу нахождения площади окружности $S=2\pi R^2$, реализовав ее в калькуляторе таким вот способом:

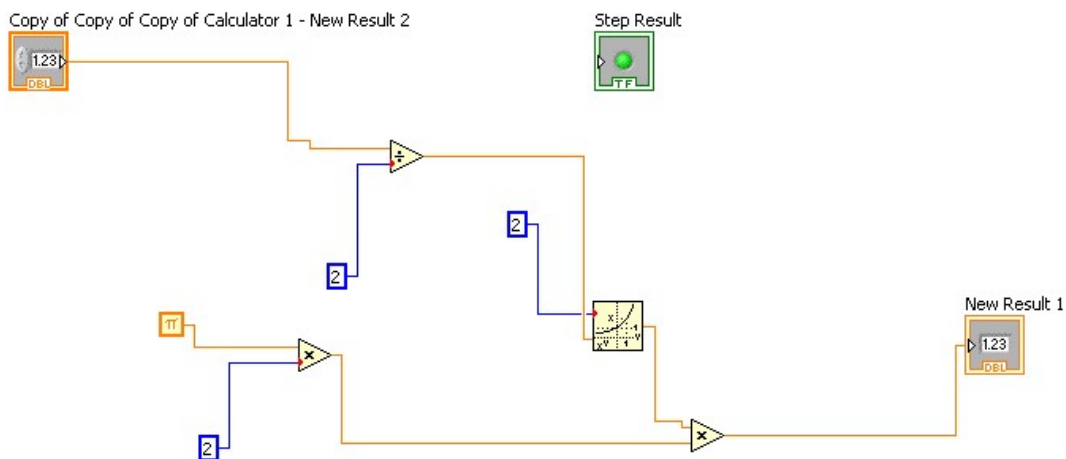


Рис.2 Реализация формулы в калькуляторе

Входную величину делим на 2, ввиду того что ее берем из калькулятора в мм, что бы получить результат так же сразу в мм, но в предыдущем калькуляторе был найден диаметр потому мы и поделили.

Проделав все это мы получим программу способную распознать любую из трех фигур (квадрат, треугольник или круг) и в добавок способную измерить нужные нам величины, сохраните результат вашей работы и попробуйте разные комбинации образцов.

Работа №4 Комплексная система распознавания объектов

Задание: спроектировать управляющую программу распознающую как отдельно образцы фигур(квадрат, треугольник и круг) так и одновременно находящиеся на одном образце на объектном -ориентированном языке программирования в среде NI Vision builder AI v. 3.6.

Подобная задача несколько усложнит имевшийся у нас до этого алгоритм, а именно добавится дополнительная ветвь с расчетами для случая, когда на образце более одного объекта. Алгоритм примет вид:

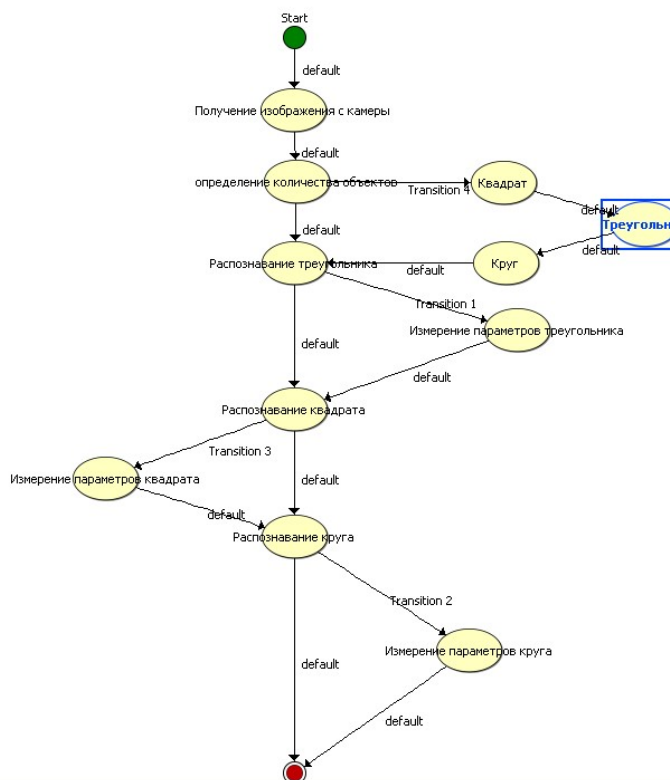


Рис.1 Алгоритм управляющей программы

Как можно заметить появился ряд новых блоков «определение количества объектов», «круг», «квадрат» и «треугольник».

В блоке определения количества объектов мы используем функцию Detect Objects-очень удобная функция когда не требуется классифицировать объекты, а нужно лишь их подсчитать панель настройки выглядит так:

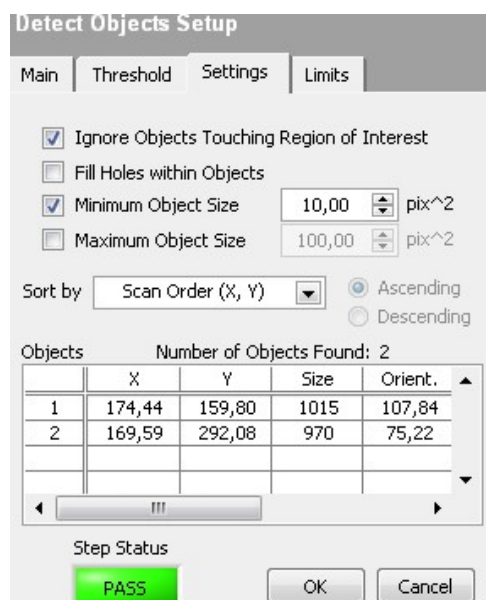


Рис.2 Окно настройки функции

Тут можно настроить минимальные и максимальные размеры детектируемых объектов, что довольно удобно ввиду того что камера способна различать самые мелкие дефекты на образцах. Так же в блоке присутствует логический калькулятор, который нужно настроить на поиск 3 объектов, если он их находит то идет выполнять измерения по пути 3х фигур на одном образце, иначе идет прямо и ищет одну фигуру, условия соответствует имеющимся в наличии образцами.

В блоках с наименованиями фигур мы реализуем систему функций на подобие той, что были ранее, только исключая блоки: 2ое использование блока линейных измерений(в данном задании достаточно найти длинны сторон и площади фигур) и соответственно не используем повторно калькулятор перевода пикселей в миллиметры.

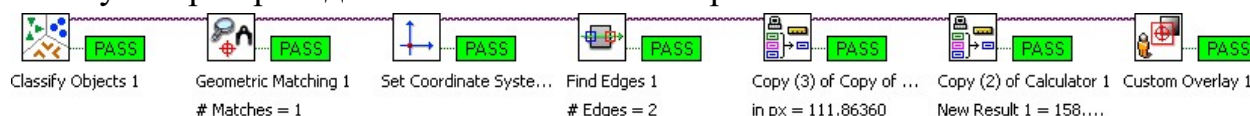


Рис. 3 Вид функциональной последовательности

В каждом из блоков с наименованием фигур следует привязать блок Geometric Matching к соответствующей фигуре. Проблем с выполнением задания возникнуть не должно, если все предыдущие работы вы успешно завершили. Не забудьте сохранить написанную вами программу.

По завершению написания программы необходимо представит отчет о проделанной работе. В него должны войти:

- Составленный алгоритм.
- Иллюстрации проделанных вами действий (скриншоты с описаниями к ним)

-Вывод о проделанной работе.

Список литературы:

1. Измерения в LabVIEW/ Баран Е.Д., Морозов Ю.В. - Новосиб.: НГТУ, 2010. - 162 с.: ISBN 978-5-7782-1428-6 – <http://znanium.com/catalog/product/546030>
1. Визильтер Ю.В. Обработка и анализ цифровых изображений с примерами на Lab VIEW IMAQ Vision.-М.:ДМК Пресс, 2007.-467 с.
2. Гонсалес Р. Цифровая обработка изображений.- М.:Техносфера, 2005
3. Гаскаров Д.В. Интеллектуальные информационные системы. Учеб. для вузов. – М.: Высш шк., 2003.- 431 с.
4. Павлов А.Н. Интеллектуальные средства измерения : учеб. пособию- Бийск : АлтГТУ, 2011.-76 с.
5. Раннев Г. Интеллектуальные средства измерений. – М.:Академия, 2011.-272 с.