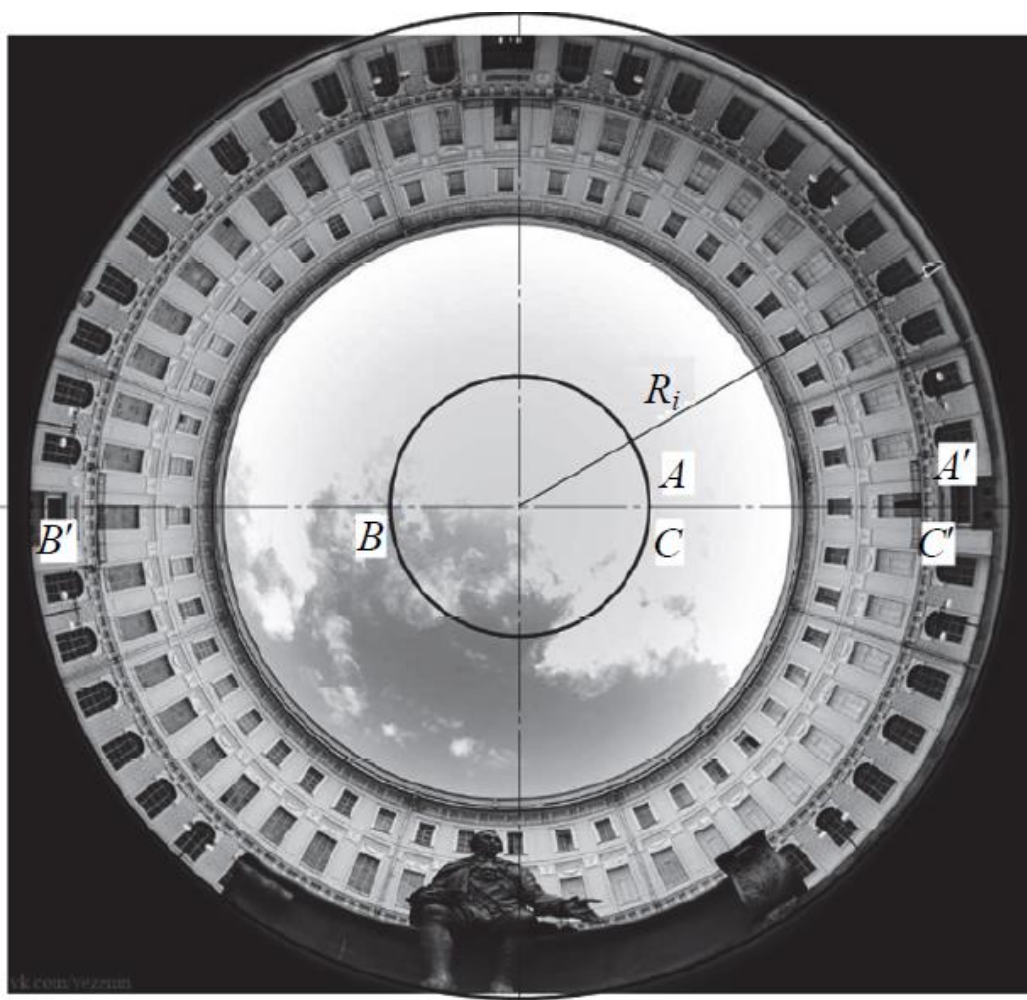


С.Н. Ярышев, Е.А. Сычева, А.А. Горбачёв

**Видеосистемы безопасности.
Методические указания по выполнению
лабораторных работ**



**Санкт-Петербург
2019**

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

С.Н. Ярышев, Е.А. Сычева, А.А. Горбачёв

**Видеосистемы безопасности.
Методические указания по выполнению
лабораторных работ**

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки 12.04.02 «Оптотехника» в качестве учебного
пособия для реализации основных профессиональных образовательных
программ высшего образования магистратуры

 УНИВЕРСИТЕТ ИТМО

Санкт-Петербург
2019

Ярышев С.Н., Сычева Е.А., Горбачев А.А. Видеосистемы безопасности. Методические указания по выполнению лабораторных работ Учебно-методическое пособие. – СПб: Университет ИТМО, 2019. – 71 с.

Рецензенты:

Голушко М.Н., к.т.н., начальник лаборатории ООО «ЭВС»

Учебно-методическое пособие предназначено для студентов по направлению подготовки 12.04.02 «Оптотехника» в качестве учебного пособия для выполнения лабораторных работ по курсу «Видеосистемы безопасности».



Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2019

©С.Н. Ярышев, Е.А. Сычева, А.А. Горбачев., 2019

Содержание

Введение. Универсальный лабораторный комплекс для проведения лабораторных работ по цифровой видеотехнике, обработке изображений и видеоаналитике.....	4
1 Лабораторная работа «Изучение методов работы с видеопотоками и изображениями, полученными от источников видеосигнала»	10
2 Лабораторная работа «Изучение методов сжатия цифрового видеосигнала на основе дискретного косинусного преобразования»	25
3 Лабораторная работа «Изучение методов построения панорам»	35
4 Лабораторная работа «Исследование алгоритмов улучшения субъективного восприятия изображений».....	49
5 Лабораторная работа «Изучение телевизионной системы безопасности» ..	57
Список используемых источников	71

Введение. Универсальный лабораторный комплекс для проведения лабораторных работ по цифровой видеотехнике, обработке изображений и видеоаналитике

Курс «Видеосистемы безопасности», читаемый в рамках направления подготовки 12.04.02 «Оптотехника» направлен на изучение вопросов проектирования аналоговых и цифровых систем безопасности, а также методов обработки изображений и видеоинформации в них.

Цель учебного пособия подготовка обучающихся к практическому применению полученных в ходе лекционного курса знаний в области проектирования оптико-электронных видеосистем безопасности.

В результате выполнения лабораторных работ обучающиеся должны получить необходимые практические навыки, научиться творчески мыслить, делать заключения и выводы из полученных результатов.

В результате освоения материалов пособия и выполнения лабораторных работ обучающийся приобретает следующие профессиональные компетенции, а также умения и навыки:

- умение использовать современные методы исследования;
- навыки представления результатов проделанной работы;
- навыки настройки элементов цифровых телевизионных систем безопасности;
- умение работать с цифровыми камерами;
- умение использовать компьютерные технологии при выборе элементной базы или унифицированных узлов проектируемых приборов.

Все лабораторные работы выполняются на специально разработанном универсальном лабораторном комплексе.

Лабораторные работы 1 – 4 направлены на изучение методов цифровой обработки изображений и видеосигналов, содержит описание основных инструментов и методов, необходимых для понимания процессов, происходящих в видеосистемах. Лабораторная работа 5 направлена на проработку навыков проектирования систем безопасности, настройки оборудования и определения основных характеристик системы.

Универсальный лабораторный комплекс (УЛК) предназначен для проведения лабораторных работ по цифровой видеотехнике, обработке изображений и видеоаналитике, а также по смежным дисциплинам, разработан и производится ООО «ЭВС» (г. Санкт-Петербург). УЛК включает в себя базовый набор, позволяющий проводить несколько лабораторных работ, наиболее характерных для цифровой видеотехники. Кроме того, имеются комплекты расширения базового набора для увеличения количества лабораторных работ и большего охвата различных областей применения изучаемых систем [1].

Основу аппаратной части лабораторной установки составляет персональный компьютер с одной или двумя камерами высокого

разрешения (рис. 1), подключенными по интерфейсу USB. В составе установки имеется:

- оптический стол;
- стойки для закрепления камер (2 шт);
- стойка для закрепления трафарета;
- микрометрическая подвижка на стойке с источником излучения;
- стойка с подсветкой;
- набор тестовых изображений;
- светозащитный кожух;
- блок питания излучателей.



Рисунок 1 – Внешний вид лабораторной установки

Программное обеспечение базовой лабораторной установки включает в себя:

- ОС Windows 7 или выше;
- драйверы камер USB;
- ПО «OSC 16»;
- ПО «Тайфун»;
- ПО Matlab;
- ПО LabView;
- файлы ПО и описаний базового комплекта лабораторных работ;
- файлы и описания быстрого освоения методов интеграций камер в собственные проекты на базе Matlab и LabView.

Расширенный состав УЛК включает:

- дополнительные IP-камеры и сетевой коммутатор с PoE, что позволяет изучить сетевые IP-камеру и сетевую распределенную систему телевизионной безопасности;

- дополнительную аналоговую камеру и аналоговую плату видеозахвата, что позволяет изучить режимы управления чувствительностью и аналоговую телевизионную систему безопасности;
- трехкоординатную микрометрическую подвижку с возможностью вертикальной установки камер, что позволяет изучать алгоритмы обнаружения и селекции объектов в системах технического зрения.

Основные технические характеристики УЛК приведены в таблице 1.

Таблица 1 – Основные технические характеристики УЛК

	Базовый вариант	Расширенный вариант
Количество телевизионных камер	2	5 и более
Формат изображения камер	5 МП	5МП, D1
Способы ввода видеосигнала	USB	USB, Ethernet, CCIR
Подсветка	Светодиодная многоцветная	Светодиодная многоцветная
Используемый компьютер	Core I5	Core I7
Используемый монитор	1920×1080, 24”	1920×1080 или 3840×2160, 24” и более
Габаритные размеры (без учета компьютера), мм ДхШхВ	1200×300×300	1200×300×500 (300)
Потребляемая мощность (без учета компьютера)	40 Вт	40 Вт
Масса (без учета компьютера), кг	25	30

Основные элементы лабораторной установки: измерительная телевизионная камера VEC-545 с интерфейсом USB, IP-камера VEC-556-IP-N (С PoE), высокочувствительная камера с ночным режимом VNC-748-N3 (рис. 2 а), линейная микрометрическая подвижка с диапазоном перемещения 25 мм и погрешностью установки 0,01 мм (рис. 2 б), угловая микрометрическая подвижка с угловым диапазоном 360° (рис. 2 в, приведена в комплекте с двумя ортогональными линейными подвижками), стойка с трафаретом испытательной таблицы (рис. 2 г).



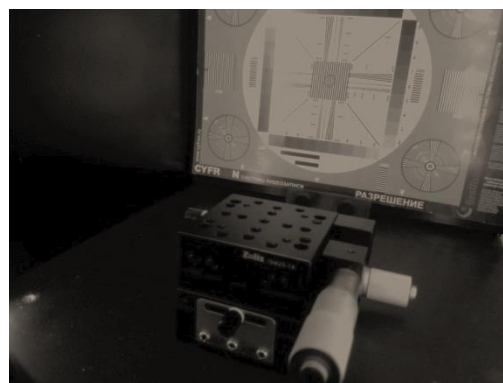
а



б





в






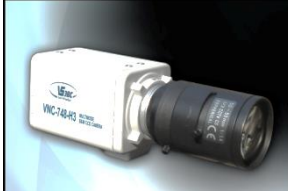
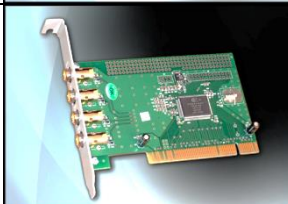



г

Рисунок 2 – Основные элементы универсальной лабораторной установки

Таблица 2 – Состав универсального лабораторного комплекса

Название оборудования	Изображение	Кол-во
Персональный компьютер (с монитором не менее 24 дюймов)		1
Камера USB (VEC-545-USB)		2
Оптическая скамья алюминиевая размер 1200×300×12 [2]		1
Дополнительная плата 300×250×12 [3]		
Микрометрическая подвижка 50 мм [4]		1

Держатель трафарета [5]		1
Стойка монтажная [6]		10
Осветитель лабораторный ОИ-19 [7]		1
Светозащитный кожух. Собственного изготовления. Размер соответствует оптической скамье 1200×300×300. По бокам ручки или открывающаяся крышка		1
Расширенный состав		
IP-камера (VEC-556-IP-N (С PoE))		2
Сетевой коммутатор с PoE 5 портов [8]		1
Аналоговая камера с ночными режимам. VNC-748-N3 с управлением по RS-232		1
Плата видеозахвата аналоговая. Цунами-4		1

Трехкоординатная микрометрическая подвижка [9]		1
Стойка для вертикальной установки камер [10]		1

Общие методические рекомендации для выполнения представленных в пособии лабораторных работ:

- лабораторные работы выполняются строго в соответствии с описанием, представленным для каждой работы.
- допуск к выполнению работы дает преподаватель или лаборант.
- для подготовки к лабораторным работам рекомендуется ознакомиться с теоретической частью, в соответствии с тематикой работы.

1 Лабораторная работа «Изучение методов работы с видеопотоками и изображениями, полученными от источников видеосигнала»

Цель работы

Изучение методов получения видеопотоков и изображений от реальных источников видеосигнала и основных методов обработки этих сигналов с использованием пакета Matlab. Для работы требуются базовые знания по пакету Matlab [11, 12].

Введение и основные теоретические сведения

Пакет Matlab является мощным средством разработки и изучения алгоритмов обработки сигналов различных типов. В старших версиях Matlab предусмотрена возможность обработки изображений и видеосигналов как наиболее сложных видов цифровых сигналов.

В пакете предусмотрены два специальных набора инструментов Image Processing Toolbox и Image Acquisition Toolbox, которые содержат уже в готовом виде основные алгоритмы обработки изображений и примеры их использования. Кроме того, следует помнить, что изначально пакет Matlab позиционировался как средство работы с матрицами (двумерными массивами данных), к которым можно отнести и изображения. Поэтому многие другие возможности обработки изображений размещены в других разделах Matlab.

Последние версии Matlab имеют возможность взаимодействия с физическими объектами, в частности с источниками видеосигналов. К таким источникам можно отнести большинство используемых в настоящее время источников видеосигнала:

- веб-камеры, подключаемые через интерфейс USB;
- веб-камеры, встроенные непосредственно в ноутбуки и моноблоки;
- камеры высокого разрешения, использующие интерфейс USB;
- камеры специального назначения, использующие интерфейсы IEEE1394, CameraLink и другие;
- IP-камеры, подключаемые через интерфейс Ethernet;
- источники тепловизионного сигнала с различными интерфейсами;
- источники сигнала, использующие двумерное представление в виде квазиизображений (томографы, аппараты УЗИ, рентгенографы, радары и пр.);
- приемники ТВ сигнала (ТВ-тюнеры аналоговые и цифровые, спутниковые приемники);
- телевизионные системы наблюдения и регистрации (ТСНР) различного назначения;
- источники потокового видео (веб-сервисы хранения видеоданных типа Youtube, удаленные интернет-камеры, интернет-телевидение и пр.);

- аналоговые источники видеосигнала, которые могут быть подключены посредством плат видеозахвата и видеотюнеров с различными интерфейсами.

Наиболее просто могут быть подключены источники видеосигнала, которые в системе Windows идентифицируются как видеоустройства и используют драйверы WDM (Windows Driver Model). В этом случае Matlab использует такие устройства как стандартные устройства Windows. В некоторых случаях могут возникнуть проблемы с визуализацией свойств видеоборудования, которые предоставляет драйвер. Данные проблемы могут быть вызваны с возможной несовместимостью драйвера и версии Matlab, связанной с разрядностью драйверов, версии Windows и программы Matlab.

Другой способ подключения источника видеосигнала более сложный и предполагает настройку параметров ввода видеосигналов. Например, для IP-камер потребуется вводить сетевые настройки, параметры протокола RTSP и некоторые другие параметры.

Для быстрой настройки стандартных источников можно воспользоваться встроенным в Matlab приложением Image Acquisition Tool.

Состав лабораторной установки

Универсальная лабораторная установка, включающая в себя (рис. 1):

- персональный компьютер и монитор;
- камера VEC-545 с интерфейсом USB на жестком основании;
- транспарант с установленной на нем измерительной телевизионной таблицей;
- светодиодный осветитель;
- блок питания осветителя.

Программное обеспечение включает в себя:

- операционную систему Windows7 или старше;
- драйверы камер и других устройств;
- пакет Matlab.

Ход работы

С помощью преподавателя или лаборанта включить лабораторную установку. Самим ничего не включать!

Запуск приложения Image Acquisition

В программе Matlab приложение Image Acquisition Tool можно найти следующим образом. В главном окне программы следует перейти по вкладке APPS и в ней найти ярлык Image Acquisition (Рис. 3).

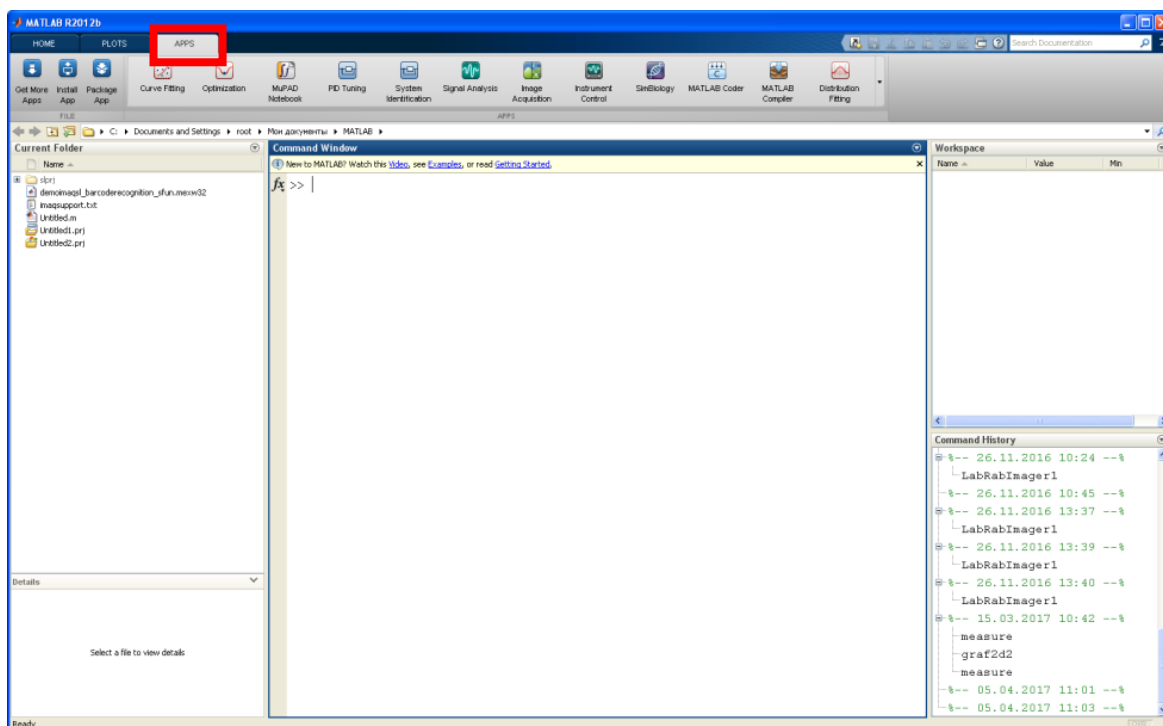


Рисунок 3 – Главное окно пакета Matlab при переходе на вкладку APPS

Запустить приложение можно щелкнув по ярлычку Image Acquisition. Приложение сначала некоторое время анализирует состав доступного оборудования и затем появляется в виде окна, которое показано на рис. 4.

В окне Hardware Browser можно посмотреть список доступного в программе Matlab видеооборудования и его основные режимы работы. Нужно иметь в виду, что на рис. 4 приведен пример списка камер. В вашей работе может присутствовать другой набор камер.

Из списка оборудования не всегда понятно, какая камера в списке соответствует реальной, особенно если подключено к компьютеру несколько одинаковых камер. Для определения камеры следует выбрать один из режимов работы и нажать кнопку Start Preview. В центральном окне вы увидите видео от выбранной камеры в выбранном режиме (рис. 5). На этом и последующих рисунках приведены примеры изображений камер, которые могут отличаться от изображений, которые вы получите на лабораторной установке.

В левом нижнем окне вы можете видеть фрагмент кода Matlab, с помощью которого осуществляется подключение камеры, выбор режима и запуск режима Preview непосредственно в тексте программы Matlab, которые можно использовать либо в командном режиме либо вставить в m-файл.

Создайте в программе Matlab новый m-файл и скопируйте туда фрагмент текста из правого нижнего окна. Запустите полученный m-файл. Вы должны получить окно с видео от выбранной камеры. Закройте это окно.

Выберите по очереди все доступные режимы работы всех доступных камер. Запускайте по очереди Start Preview для каждого режима.

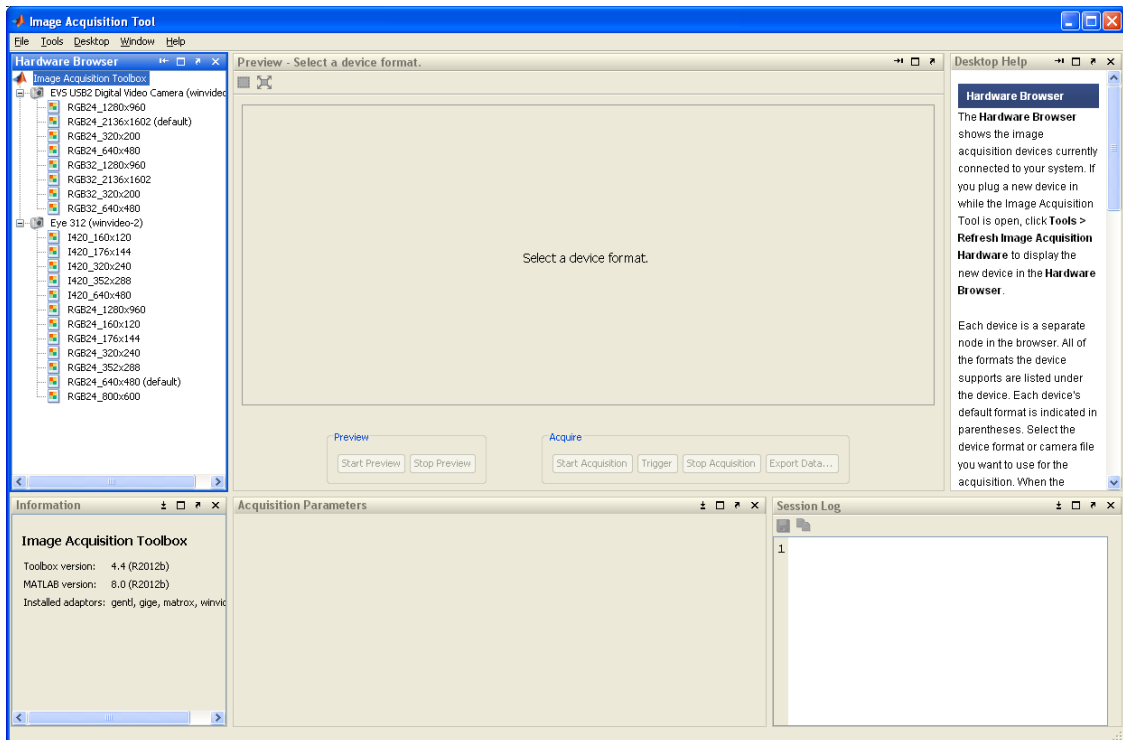


Рисунок 4 – Окно приложения Image Acquisition Tool

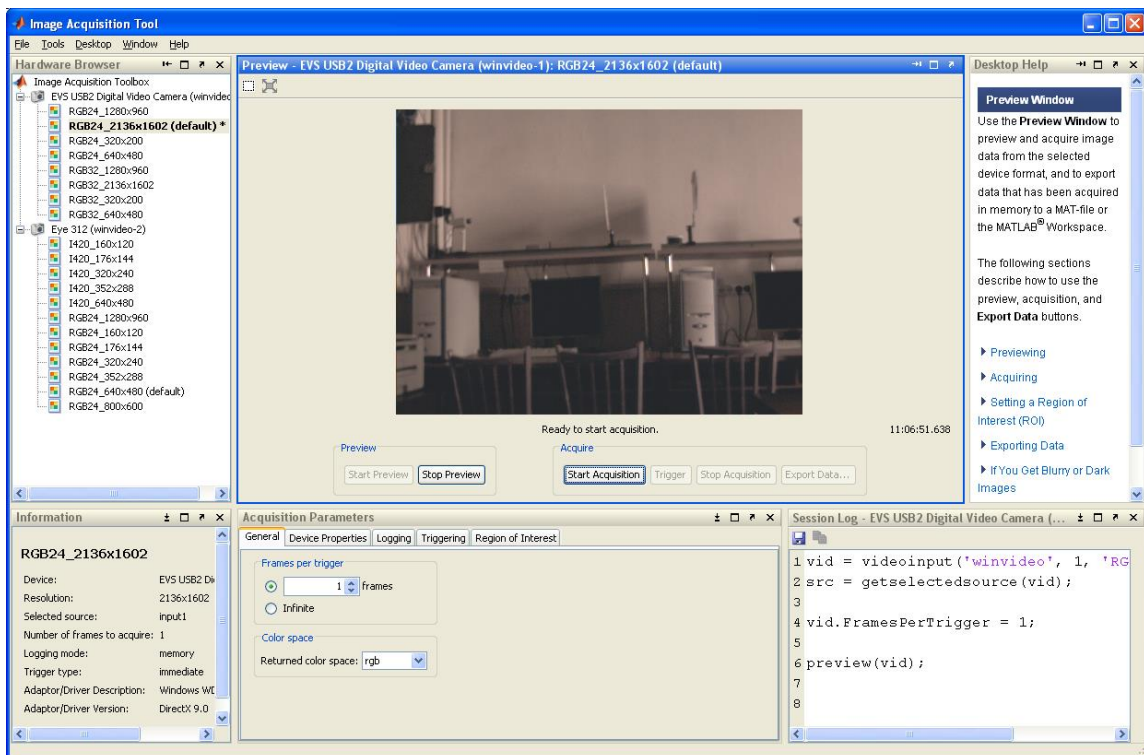


Рисунок 5 – Окно видео после нажатия кнопки Start Preview в окне приложения Image Acquisition Tool

Выбор параметров камеры

Выберите в списке режимов камеру, которая соответствует реальной камере, у которой в поле зрения установлена испытательная таблица. У этой камеры выберите наибольший формат изображения. Запустите режим Preview. Обратите внимание на нижнее среднее окно приложения Image Acquisition Tool – Acquisition Parameters (рис. 6). В этом окне есть несколько вкладок. В первой из них можно изменить метод представления цвета – RGB, YCbCr или Grayscale. Здесь оставьте режим неизменным.

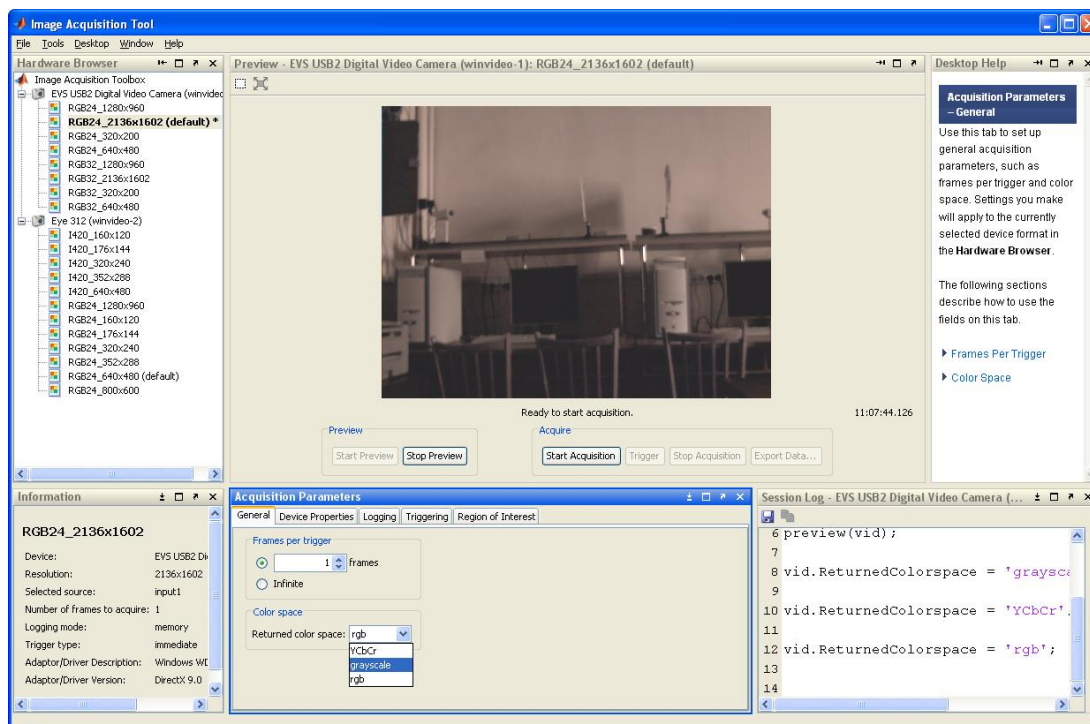


Рисунок 6 – Выбор нужного режима камеры в приложении Image Acquisition Tool

В окне Acquisition Parameters следует перейти во вторую вкладку Device Properties. Вы увидите список доступных регулируемых параметров и режимов и их значения в настоящий момент (рис. 7). Для удобства можно изменить соотношение между верхней и нижней частью приложения Image Acquisition Tool.

Последовательно выполните регулировку режимов камеры, передвигая движок ползунка в положение, соответствующее наилучшему изображению. После всех регулировок перейдите в правое нижнее окно и скопируйте тест в окно открытого m-файла (рис. 8).

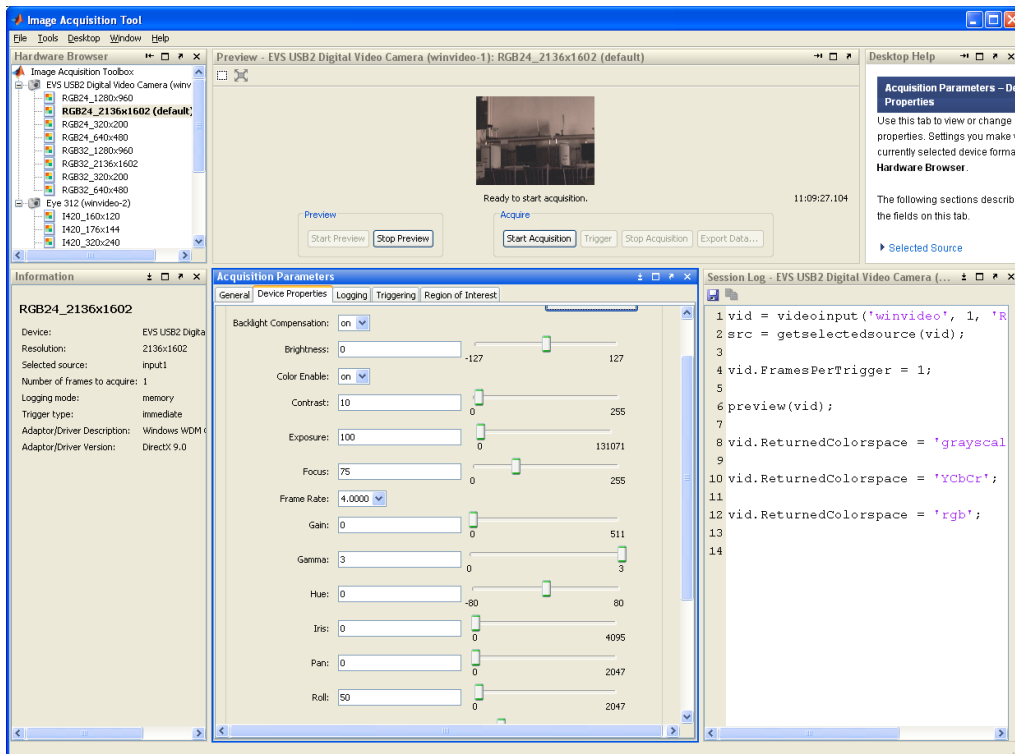


Рисунок 7 – Список доступных регулируемых параметров во вкладке Device Properties

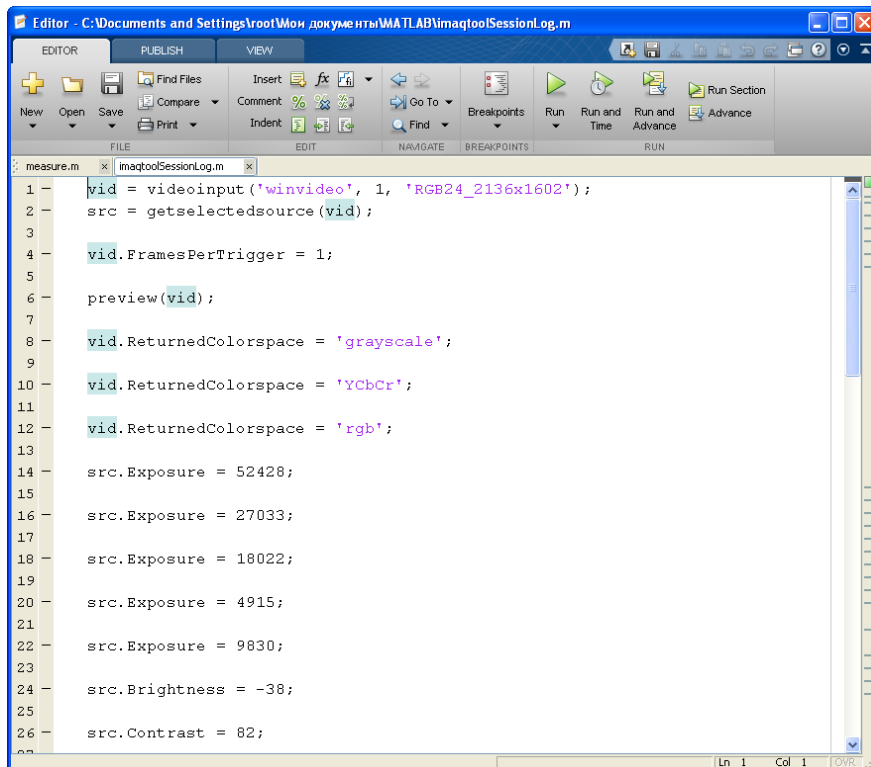


Рисунок 8 – Внешний вид открытого m-файла со скопированным в него фрагментом настроек, полученных в ходе работы Image Acquisition Tool

Запустите полученный m-файл. Вы должны увидеть окно Preview с камеры, у которой выполнены настройки параметров. Теперь эти настройки выполняются в ходе выполнения скрипта m-файла.

В ходе работы с изображениями и видеопотоками может пригодиться выбор региона интереса в изображении. Перейдите к вкладке Region of Interest в окне Acquisition Parameters. В окне вы увидите условное изображение окна, в котором в последующем вы будете работать. Вверху вы должны видеть режим просмотра Preview (при необходимости следует нажать кнопку Start Preview).

В окне Preview можно мышкой выбрать регион интереса в виде прямоугольного окна (рис. 9). Затем нажмите кнопку Select or Edit. Вы увидите в окне Preview изображение, которое соответствует выбранному окну. Кроме того, вы в нижнем окне увидите четыре значения координат, ограничивающих выбранный регион интереса (рис. 10), а в левом нижнем окне вы увидите команды программы Matlab, которые соответствуют выбору региона интереса.

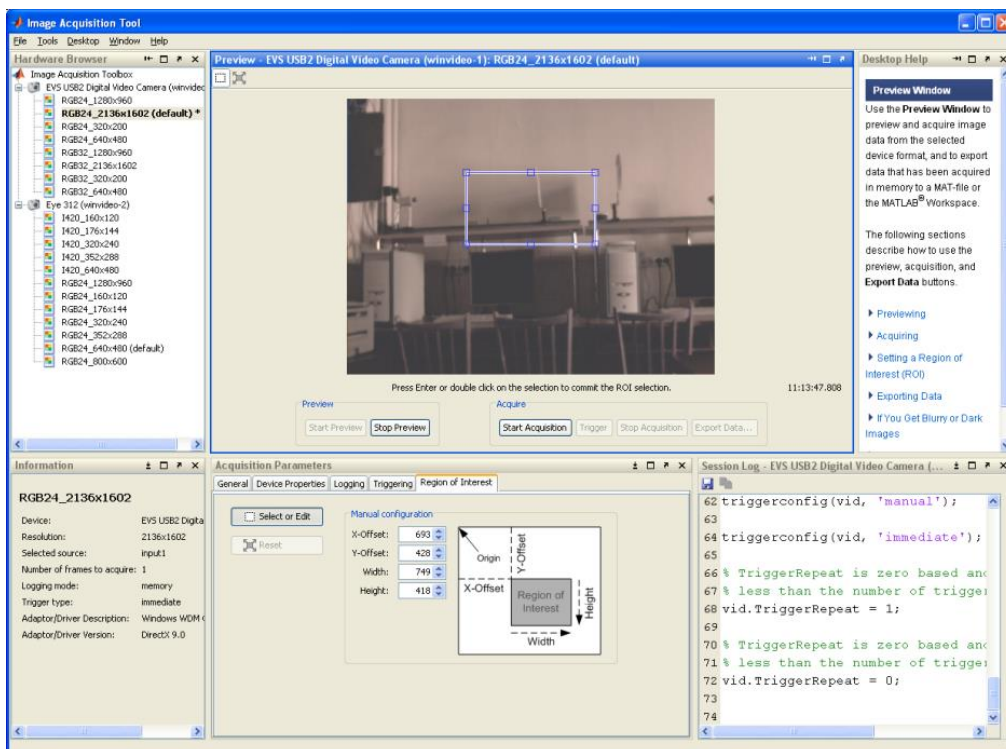


Рисунок 9 – Подготовка режима Выбора региона интереса

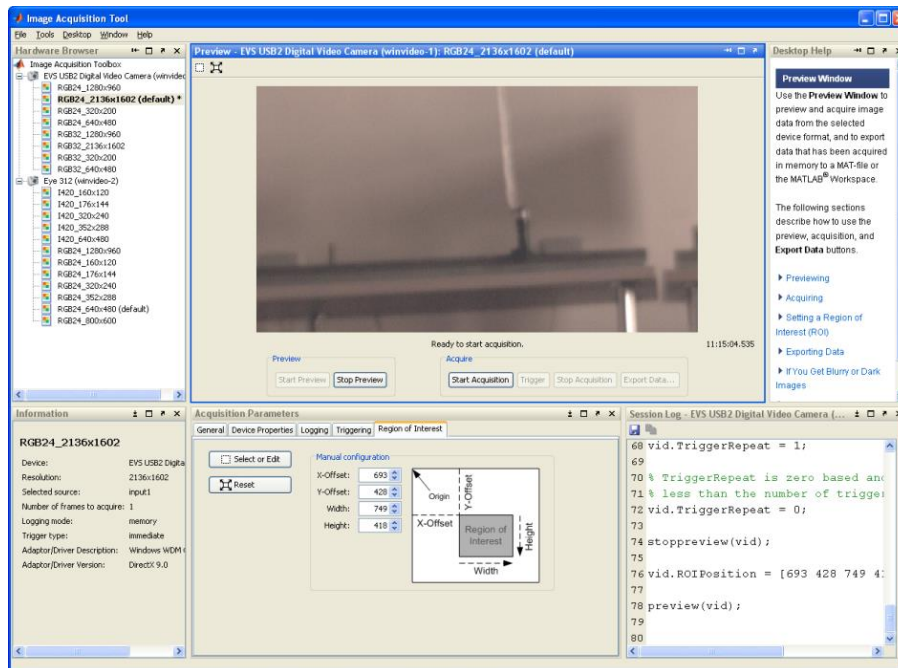


Рисунок 10 – Запуск режима Выбора региона интереса

Все выполненные вами действия отображаются в виде команд в правом нижнем окне приложения Image Acquisition Tool. Для сохранения последовательности команд можно не только копировать их в открытое окно m-файла, но и сохранить целиком все ваши действия, которые вы выполняли во время настройки камер в виде отдельного m-файла. Для этого следует нажать на изображении дискеты в правом нижнем окне Image Acquisition Tool. Открывшееся окно предложит сохранить весь текст из правого нижнего окна Image Acquisition Tool в m-файле с выбранным названием (рис. 11). Все ваши действия будут сохранены в виде этого m-файла. Но, перед тем как выполнить это действие, лучше просмотреть все команды в правом нижнем окне Image Acquisition Tool и удалить из него ненужные команды.

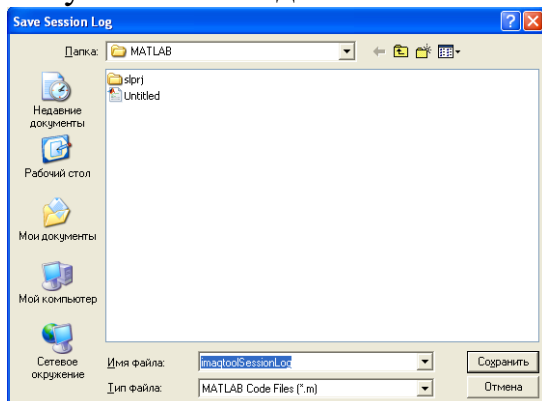


Рисунок 11 – Сохранение параметров и настроек Image Acquisition Tool в виде m-файла

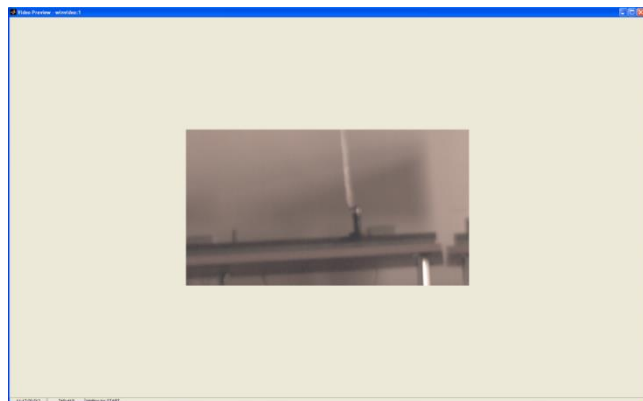


Рисунок 12 – Результат запуска m-файла с сохраненными настройками камеры

Использование настроек Image Acquisition Tool в основной программе Matlab

Запустите сохраненный m-файл. Вы должны увидеть окно с Preview от выбранной камеры, у которой были произведены настройки и выбран регион интереса (рис. 12).

Следует иметь в виду, что в случае, если ранее от той же камеры был запущен режим Preview и не остановлен (не закрыто окно Preview), то вы получите сообщение об ошибке в основном окне программы Matlab о том, что это видеопоток уже занят (рис. 13). В этом случае старое окно Preview следует закрыть, тем самым остановив видеопоток от данной камеры.

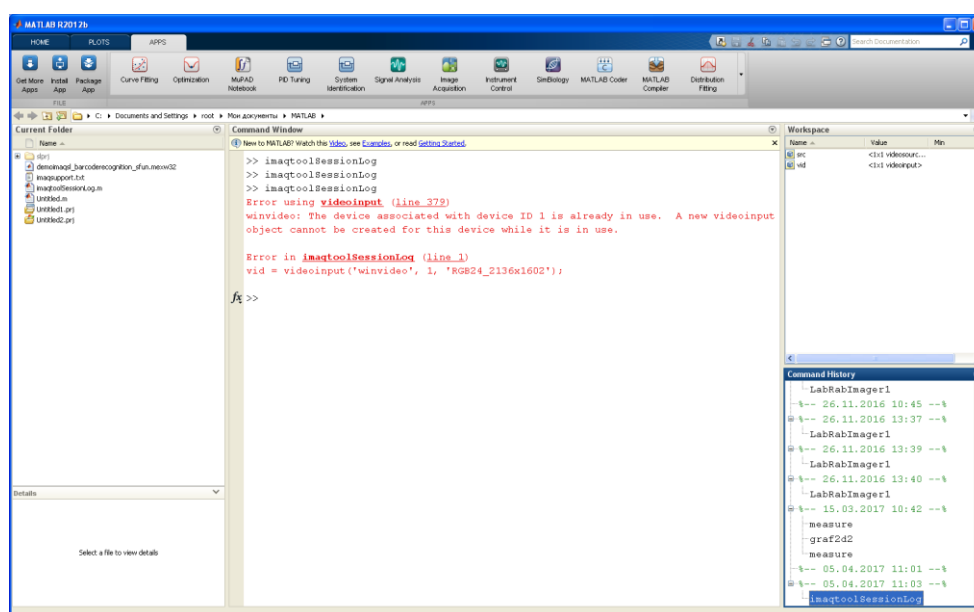


Рисунок 13 – Сообщение об ошибке при попытке обращения к уже открытому видеопотоку

Работа одновременно с несколькими видеоисточниками

Этот пункт выполняется только в том случае, если к компьютеру подключено две или более видеокамеры.

Вновь запустите Image Acquisition Tool и выберите из списка полученных камер один из режимов с максимальным разрешением. Запустите режим Preview. Убедитесь, что вы получаете видеопоток. Затем перейдите к спискам режимов второй камеры и также выберите один из режимов с максимальным разрешением. Запустите Preview и убедитесь, что теперь вы видите изображение уже с другой камеры.

Скопируйте скрипт из правого нижнего окна в m-файл любым уже известным вам способом. Вы получите окно m-файла примерно как на рис.14.

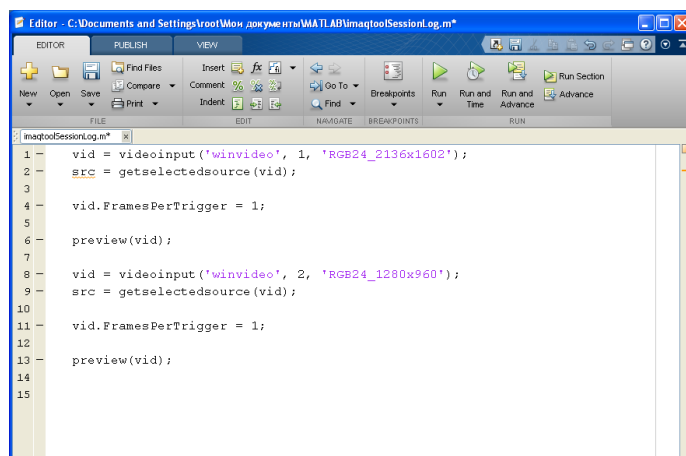


Рисунок 14 – Окно с настройками и запуском сразу двух видеопотоков с двух камер

При попытке запуска такого m-файла возможно появление ошибки в главном окне Matlab (рис. 15).

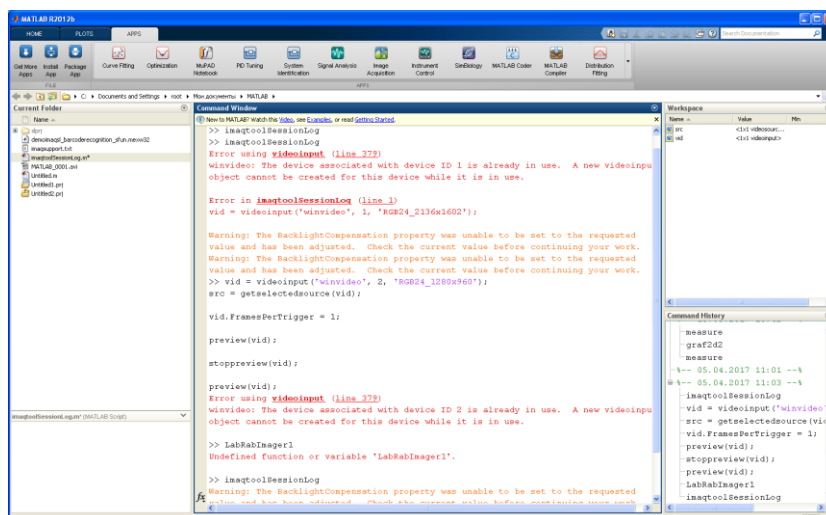


Рисунок 15 – Ошибка, возникающая в окне Matlab при попытке использовать одну переменную видеопотока сразу для двух камер

Смысл этой ошибки заключается в том, что для обеих камер в скрипте используется одна и та же переменная видеопотока vid. Поэтому при запуске скрипта обнаруживается попытка использования одного и того же видеопотока для обеих камер.

Отредактируйте m-файл вручную таким образом, чтобы для первой камеры использовался видеопоток vid, а для второй камеры видеопоток vid1 (рис. 16). Аналогично, для второй камеры переименуйте имя источника (камеры) на src1.

```

1 vid = videoinput('winvideo', 1, 'RGB24_2136x1602');
2 src = getselectedsource(vid);
3
4 vid.FrameTrigger = 1;
5
6 preview(vid);
7
8 vid1 = videoinput('winvideo', 2, 'RGB24_1280x960');
9 src1 = getselectedsource(vid1);
10
11 vid1.FrameTrigger = 1;
12
13 preview(vid1);
14
15

```

Рисунок 16 – Исправленный m-файл для двух камер

Запустите исправленный m-файл. На экране вы должны увидеть два запущенных окна Preview, которые соответствуют двум камерам (рис. 17).

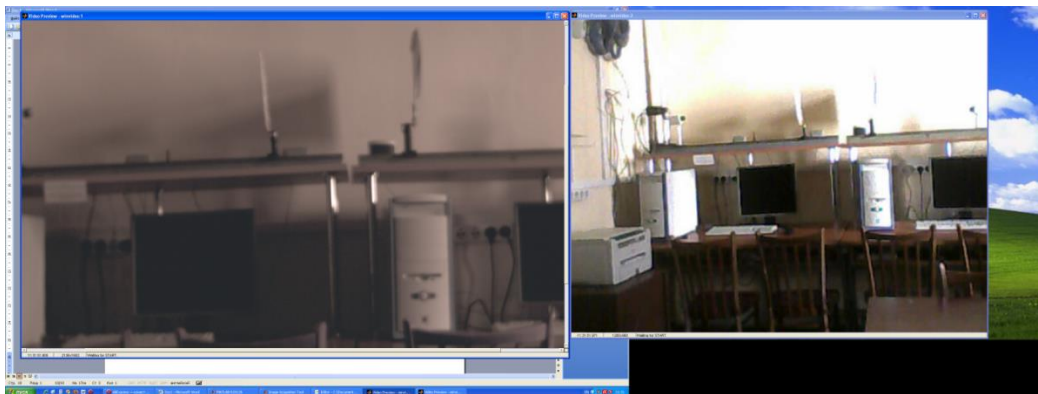


Рисунок 17 – Окна Preview для двух используемых камер

Обратите внимание на окно Workspace в правом верхнем углу программы Matlab (рис. 18). Там имеется информация обо всех переменных, которые создаются и используются по мере создания и запуска программ.

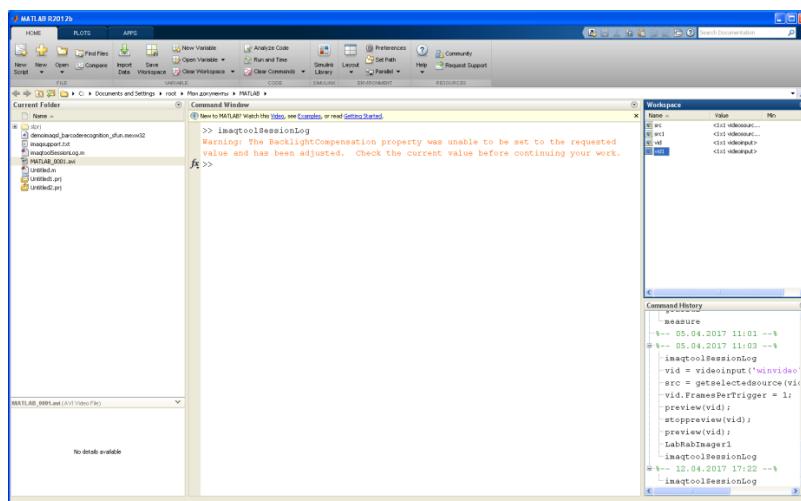
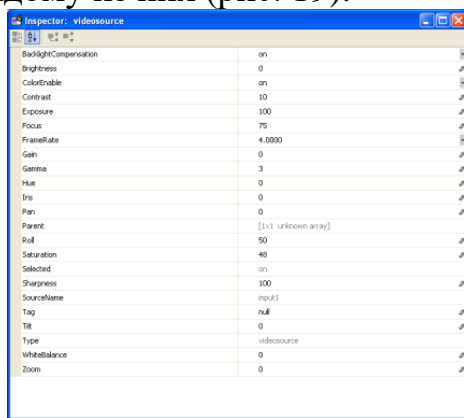
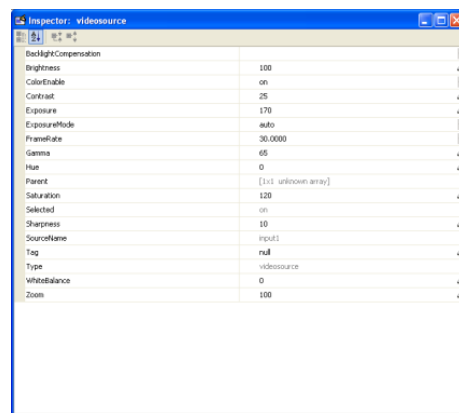


Рисунок 18 – Положение окна Workspace в программе Matlab

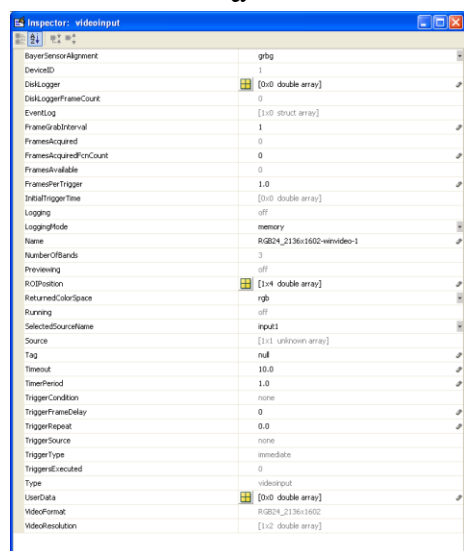
Для нашего случая в этом окне будут сведения о двух используемых видеисточниках (камерах) src и src1 и двух видеопотоках vid и vid1. Информацию об этих объектах можно получить, щелкнув мышкой по каждому из них (рис. 19).



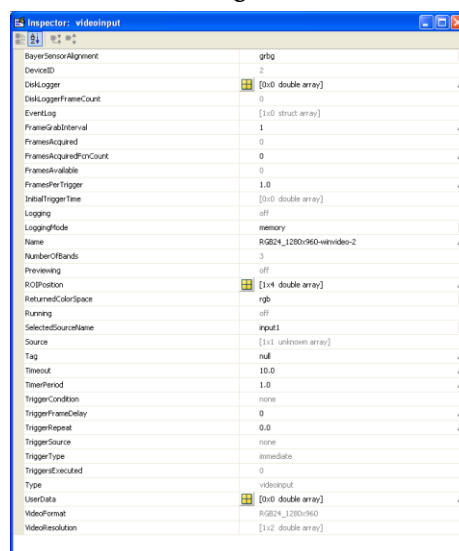
а



б



в



г

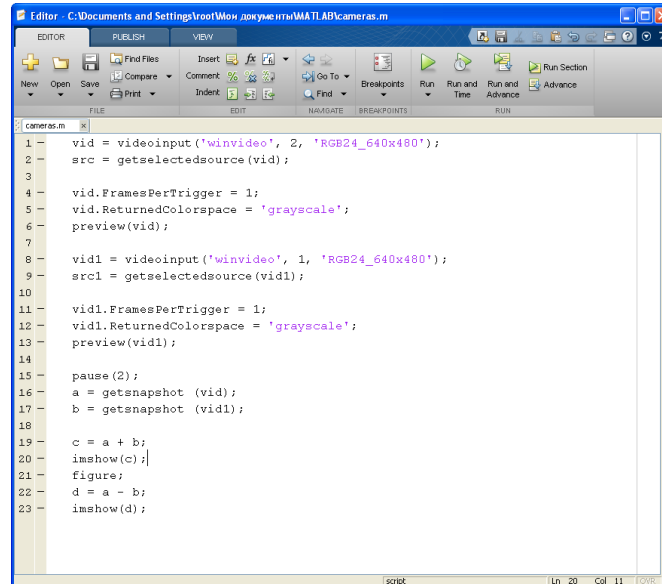
Рисунок 19 – Информация об объектах src и src1 (а, б), vid и vid1 (в, г)

Совместная обработка изображений от двух источников

С помощью приложения Image Acquisition Tool создайте скрипты для обеих камер при одинаковых настройках и занесите их в открытый m-файл (рис. 20). Например, в нашем случае это будут режимы 'RGB24_640×480'. Кроме того, для каждой камеры выберите черно-белый режим 'grayscale'. Как уже отмечалось ранее, придется вручную переименовать один из видеисточников и видеопотоков. Далее по ходу выполнения программы следует предусмотреть небольшую задержку (2 секунды), которая потребуется для выхода обеих камер на установившийся режим работы. Затем из обоих видеопотоков получим по одной реализации кадра, соответственно а и b для первой и второй камер. Визуализировать их нет смысла, так как они повторяют изображения из черно-белого Preview для обеих камер.

Следующими командами создадим суммарный кадр $c=a+b$ и межкадровую разность $d=a-b$. Полученные изображения c и d можно визуализировать.

Дополним m -файл соответствующими командами и запустим его.



```
1 - vid = videoinput('winvideo', 2, 'RGB24_640x480');
2 - src = getselectedsource(vid);
3
4 - vid.FramesPerTrigger = 1;
5 - vid.ReturnedColorspace = 'grayscale';
6 - preview(vid);
7
8 - vid1 = videoinput('winvideo', 1, 'RGB24_640x480');
9 - src1 = getselectedsource(vid1);
10
11 - vid1.FramesPerTrigger = 1;
12 - vid1.ReturnedColorspace = 'grayscale';
13 - preview(vid1);
14
15 - pause(2);
16 - a = getsnapshot(vid);
17 - b = getsnapshot(vid1);
18
19 - c = a + b;
20 - imshow(c);
21 - figure;
22 - d = a - b;
23 - imshow(d);
```

Рисунок 20 – m -файл, выполняющий захват изображений одновременно с двух камер, перевод изображений из цветного в черно-белое и создающий сумму двух кадров и межкадровую разность

Следует обратить внимание, что после выполнения указанного m -файла в окне *Workspace* помимо уже известных объектов src и $src1$, vid и $vid1$ появятся и вновь введенные объекты a , b , c , d . Каждый из этих объектов после перевода изображений в черно-белое представление представлены в виде двумерных матриц размерностью 640×480 (выбранный нами формат изображения). Каждый элемент представлен в виде целого однобайтового числа (рис. 22).

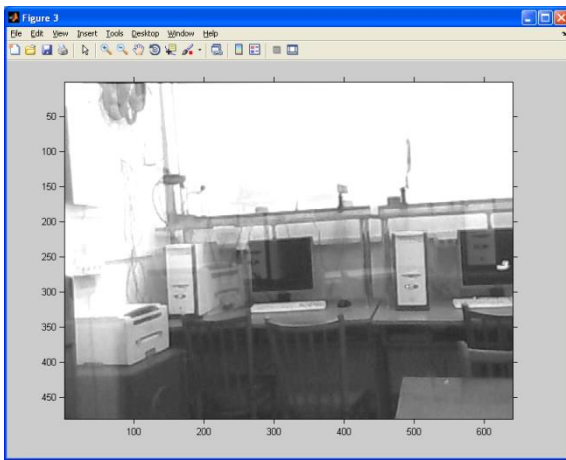
Таким образом, каждую из этих матриц в дальнейшем можно обрабатывать любыми численными методами, использующими как операции с матрицами, так и с отдельными целочисленными переменными.



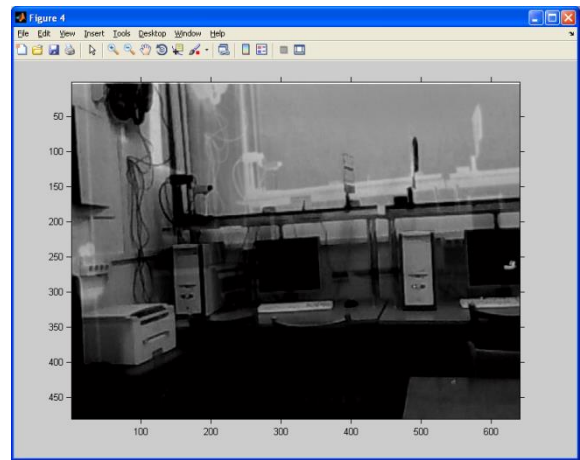
а



б



в



г

Рисунок 21 – Результат работы запущенного m-файла: а, б – изображение с камер, в – суммарный кадр, г – межкадровая разность

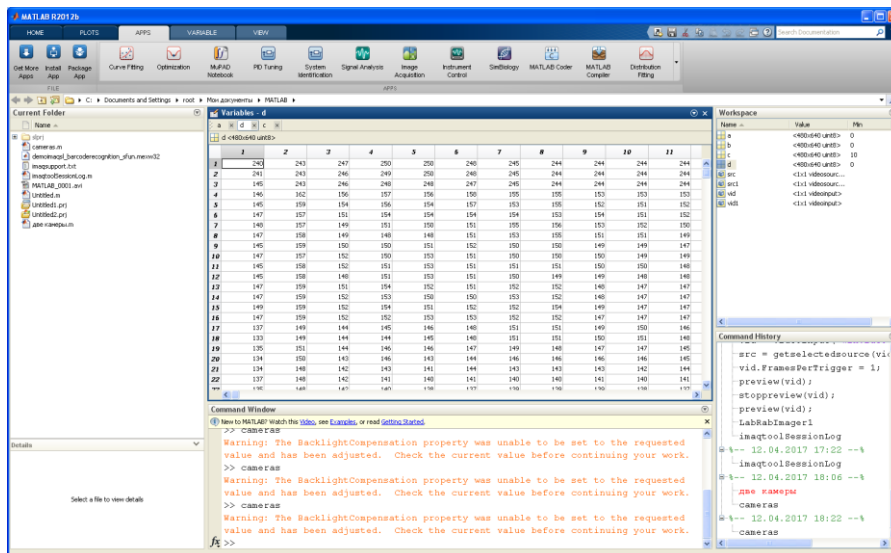


Рисунок 22 – Матричное представление одного из полученных изображений

По окончании работы самостоятельно оборудование не выключать, а обратиться к преподавателю или лаборанту.

Оформление отчета

При оформлении отчета следует пользоваться созданным шаблоном отчета. В шаблон следует занести номер группы и ФИО студентов, участвующих в работе.

В отчете должны присутствовать скриншоты, иллюстрирующие основные этапы работы.

В отчете в свободной форме следует описать порядок подключения камер в тексте программы Matlab.

Указать, какие параметры и характеристики получаемого изображения используются в программных настройках. Каким образом может быть получена двумерная матрица, соответствующая цифровому представлению изображения.

Вопросы по работе

1. Что такое пакет Matlab?
2. Для чего можно использовать приложение Image Acquisition Tool?
3. Как можно получить и использовать видеопоток с реальных камер в программах Matlab?
4. Какие камеры и источники видеосигнала можно использовать в пакете Matlab?
5. Какими параметрами камер можно управлять в Matlab?
6. Как получить изображение в виде матрицы для последующей обработки?
7. Что такое m-файл?

2 Лабораторная работа «Изучение методов сжатия цифрового видеосигнала на основе дискретного косинусного преобразования»

Цель работы

Изучение работы основных алгоритмов компрессии цифровых видеосигналов на основе дискретного косинусного преобразования (ДКП).

Основные теоретические сведения

Из теории обработки сигналов известно, что информационное содержание практически любого сигнала удобнее анализировать, рассматривая не изменение сигнала во времени, а его разложение на частотные составляющие. Анализ частотного состава позволяет отделить существенные информационные составляющие от менее значимых. Благодаря контролируемому удалению менее существенных составляющих можно уменьшить объем данных, требуемый для передачи и хранения информации об этом сигнале. При работе с изображениями и видеосигналом используется двумерное представление цифровой информации (в виде матриц). Тем самым, для реализации обработки таких двумерных сигналов используются двумерные преобразования.

Среди основных методов, реализующих пространственно-частотное преобразование, чаще всего используется дискретное косинусное преобразование.

Дискретное косинусное преобразование

Программно-технически дискретное косинусное преобразование (ДКП) чаще всего выполняется как последовательность матричных перемножений, которая определяется следующим образом [17]:

$$P_{DCT} = DCT \cdot P \cdot DCT^T,$$

где P – блок изображения (матрица элементов яркости или цветности), P_{DCT} – матрица коэффициентов ДКП после преобразования, DCT – матрица косинусного преобразования, DCT^T – транспонированная матрица косинусного преобразования.

Значения матрицы ДКП вычисляются следующим образом:

$$DCT_{i,j} = \frac{1}{\sqrt{N}}, \text{ если } i = 0.$$

$$DCT_{i,j} = \sqrt{\frac{2}{N}} \cos\left((2j+1)i \cdot \frac{\pi}{2N}\right), \text{ если } i > 0.$$

Как известно, матричное перемножение требует довольно больших затрат. И эти затраты существенно возрастают с увеличением формата матриц. Поэтому чаще всего используется блочная обработка изображения и размер используемых блоков составляет 8×8 элементов. Таким образом,

матрицы DCT и DCT^T при $N = 8$ можно заранее рассчитать и представить как постоянные величины.

Известно, что коэффициент корреляции соседних элементов типичного изображения близок к единице, т.е. велика вероятность того, что уровни сигнала соседних пикселей одинаковы. Дискретно-косинусное преобразование является средством декорреляции, которая позволяет описать блок отсчетов из 64 пикселей с использованием меньшего числа коэффициентов DCT . Сами же коэффициенты представляют собой величины (амплитуды), показывающие степень использования в фрагменте изображения соответствующих им базисных функций DCT . На рисунке 23 показано семейство функций DCT применительно к фрагменту форматом 8×8 элементов.

Если после преобразования используются все коэффициенты, то после обратного преобразования можно полностью восстановить исходный фрагмент изображения.

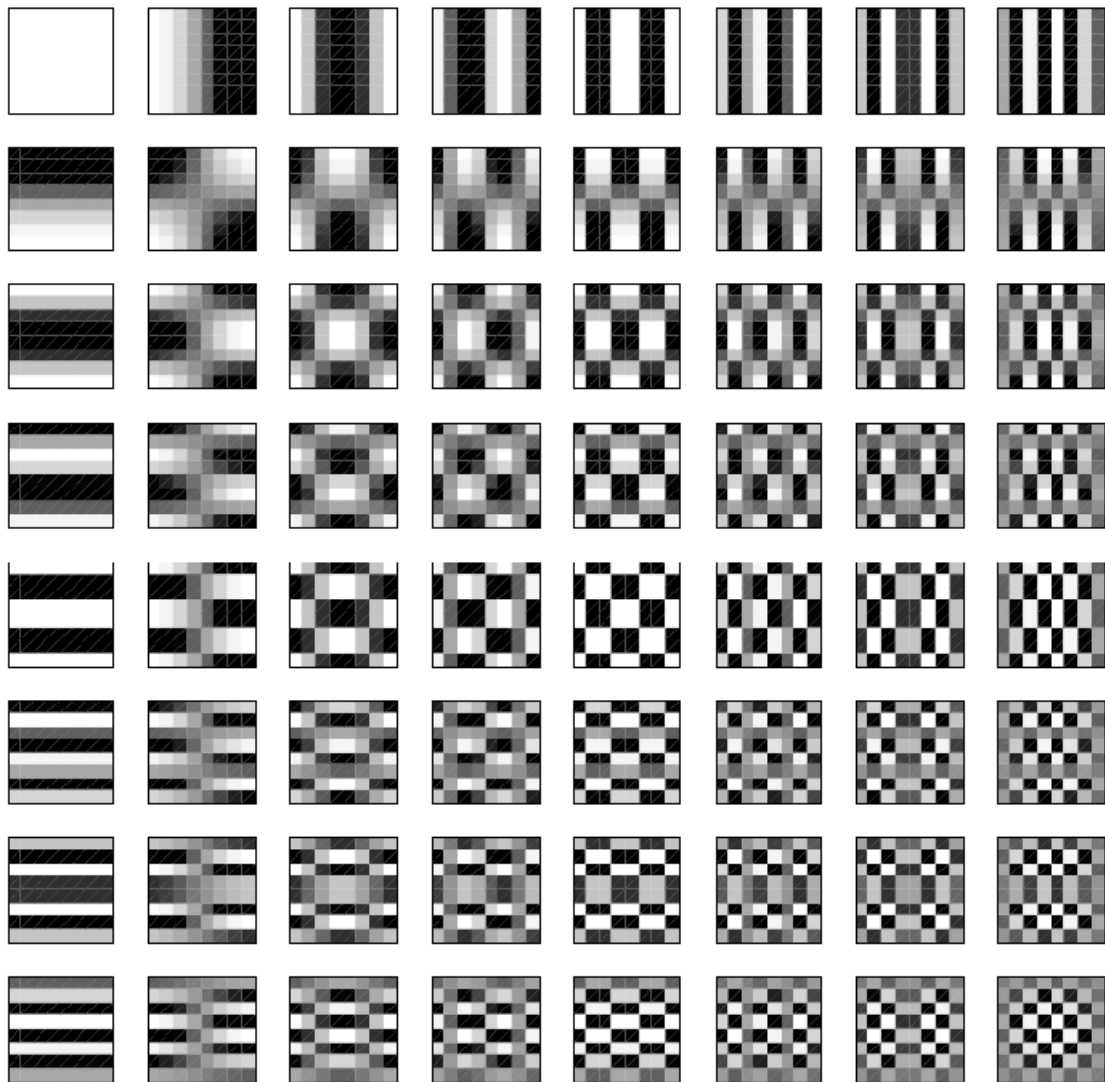


Рисунок 23 – Базовые функции ДКП

Алгоритм сжатия изображения JPEG

Алгоритм JPEG разработан специально для сжатия изображений группой экспертов в области фотографии JPEG (Joint Photographic Expert Group) и разработан на основе ДКП.

ДКП раскладывает изображение на набор коэффициентов, часть из которых может быть равна нулю вследствие неиспользования некоторых функций ДКП. Уже с использованием данного факта можно добиться некоторого сжатия данных. Однако наибольший эффект достигается при удалении части малозначимых коэффициентов (приравнивания их к нулю).

Обычно внешне матрица P_{DCT} имеет хорошо заметную особенность. Численные значения элементов матрицы быстро уменьшаются от левого верхнего угла к правому нижнему углу. Таким образом, в левом верхнем углу размещаются самые важные данные, а в правом нижнем – наименее важные. Используя это факт, можно устранить наименее значимые данные. Для этого следует провести квантование преобразованных данных.

Идея квантования заключается в том, что спектральная (частотная) информация должна превышать известный порог, чтобы составить важную часть всей информации о данном фрагменте изображения. Именно на этапе квантования происходит потеря части информации и, следовательно, потеря качества.

Квантование обычно производится на основе специальной матрицы, которая содержит делители, на которые нужно будет делить элементы ДКП. Часто используется следующий алгоритм:

$$Q(i,j) = 1 + ((1 + i + j) q),$$

где $Q(i,j)$ – матрица делителей, q – параметр качества.

Путем выбора параметра q можно управлять величинами делителей и регулировать степень сжатия. Например, при $q = 2$ получится матрица следующего вида (таблица 3):

Таблица 3 – Пример матрицы квантования

3	5	7	9	11	13	15	17
5	7	9	11	13	15	17	19
7	9	11	13	15	17	19	21
9	11	13	15	17	19	21	23
11	13	15	17	19	21	23	25
13	15	17	19	21	23	25	27
15	17	19	21	23	25	27	29
17	19	21	23	25	27	29	31

После деления 64 элементов матрицы DCT^T на элементы матрицы $Q(i,j)$ в качестве результата матрицу, у которой:

- появится большое количество дополнительных нулевых значений,

- эффект уменьшения значений от левого верхнего к правому нижнему углу будет выражен еще сильнее.

Для экономичной записи требуется изменить порядок обхода полученных значений таким образом, чтобы последовательности нулевых элементов были бы как можно длиннее. Одним из возможных способов изменения порядка обхода является способ зигзаг (рис 24).

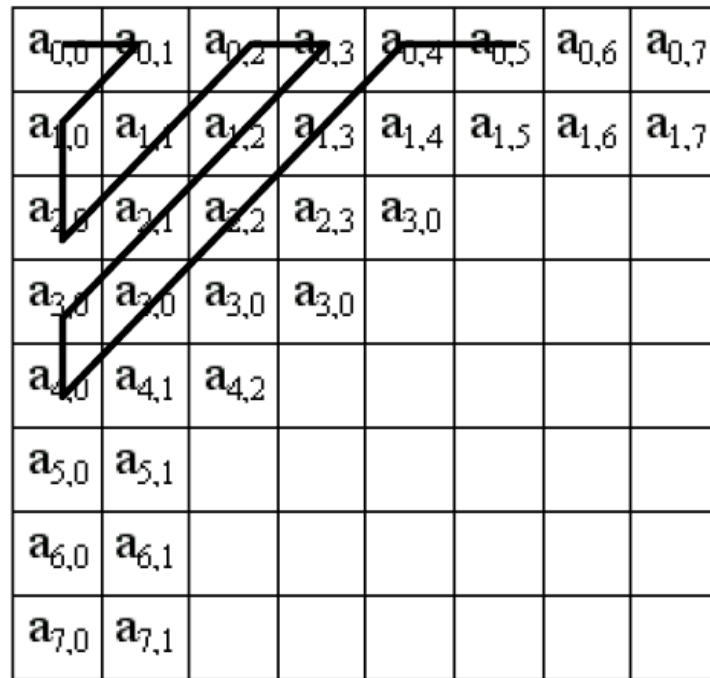


Рисунок 24 – Преобразования двумерной матрицы в одномерную последовательность по способу «зигзаг»

Как видно из рисунка, двумерная матрица форматом 8×8 элементов преобразуется в одномерную последовательность длиной 64 элемента. Главным свойством такой последовательности будет расположение наиболее значимых коэффициентов в ее начале, а наименее значимых элементов (обычно нулей) в ее конце.

Реализация алгоритма включает в себя ряд последовательных действий, который иллюстрируется на рисунке 25.

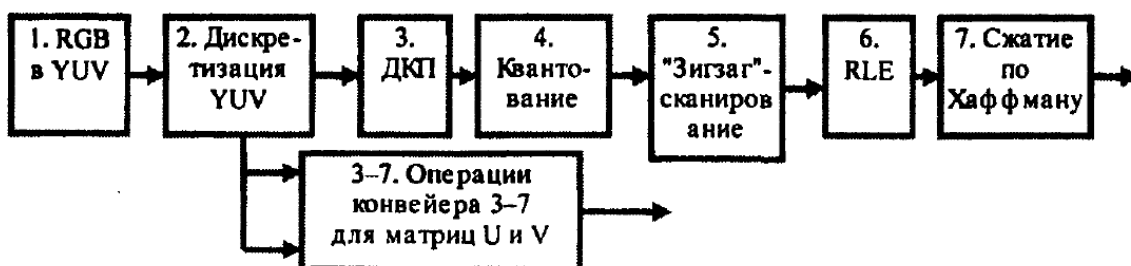


Рисунок 25 – Последовательность операций при реализации алгоритма JPEG

1. Изображение при необходимости переводится в формат YUV .
2. Производится дискретизация цветоразностных U и V сигналов в соответствии с форматом 4:2:0. Разбиение изображения на фрагменты размером 8×8 элементов. Далее обработка сигналов яркости и цветности может производиться независимо и параллельно.
3. Дискретное косинусное преобразование выполняется применительно ко всем блокам размером 8×8 элементов.
4. Квантование в соответствии с выбранным параметром качества.
5. Сканирование «зигзаг» для получения одномерной последовательности из 64 элементов.
6. Алгоритм RLE применяется к одномерной последовательности.
7. Алгоритм Хаффмана применяется к уже сжатой с помощью RLE последовательности.

П.п. 3 – 7 выполняются для всех блоков форматом 8×8 элементов и для всех цветовых плоскостей.

Основные особенности метода JPEG состоят в следующем:

- Высокий коэффициент сжатия, особенно для изображений, качество которых расценивается как хорошее или отличное.
- Большое число параметров, позволяющих пользователю экспериментировать с настройками метода и добиваться необходимого баланса сжатие/качество.
- Хорошие результаты для любых типов непрерывно-тоновых изображений независимо от их разрешения, пространства цветов, размера пикселей или других свойств.
- Достаточно изощренный метод сжатия, но не слишком сложный, позволяющий создавать соответствующие устройства и писать программы реализации метода для компьютеров большинства платформ, а также аппаратными средствами.
- Возможность использования сжатия без потерь информации при не очень высоком коэффициенте сжатия.

Описание работы

Работа проводится на персональном компьютере с установленной на нем академической версией программы Matlab. Для работы используется демонстрационный пример `dctdemo.m`, входящий в Matlab.

Для создания отчета по лабораторной работе использовать шаблон документа Word.

При запуске примера появляется окно с элементами управления. Пример использует несколько черно-белых восьмибитовых тестовых изображений форматом 128×128 пикселей. Выбрать тестовое изображение можно с помощью списка «Select Image». В левом верхнем окне

отображается исходное изображение. В правом верхнем углу приведена структура используемых коэффициентов ДКП в виде матрицы размером 8×8 . Эти коэффициенты соответствуют базовым функциям ДКП, приведенным на рисунке 23. При этом белые квадраты соответствуют коэффициентам, которые используются при кодировании изображения, черные квадраты соответствуют коэффициентам ДКП, которые не используются при кодировании, то есть приравнены к нулю вследствие операции квантования. Количество выбранных коэффициентов можно варьировать с помощью движка, расположенного под матрицей коэффициентов. Число коэффициентов можно менять от 1 (минимальное значение, соответствующее постоянной составляющей для матрицы 8×8) до 64 (используются все коэффициенты, то есть преобразование без потерь данных). Для того чтобы увидеть результат операции ДКП и квантования, после выбора числа используемых коэффициентов следует нажать кнопку «Apply». При этом в левой нижней части окна отображается изображение исходного рисунка после выполнения кодирования с использованием ДКП, операции квантования с использованием выбранного количества коэффициентов ДКП и восстановления изображения путем обратного преобразования ДКП. Степень искажений изображения будет зависеть от количества используемых коэффициентов ДКП. По этому изображению можно выполнить субъективную оценку качества изображения после компрессии. Дополнительно для оценки искажений в нижней центральной части окна приводится изображение, представляющее собой разность между исходным и восстановленным изображением. При этом пиксели, значение которых в матрицах одинаковы, отображаются как серые. Если исходное изображение и восстановленное изображение одинаковы (используются все 64 коэффициента), то преобразование выполнено без потерь и матрица ошибок представляет собой ровный серый фон. Дополнительно для объективной оценки качества восстановленного изображения в самом низу окна примера приводится вычисленная среднеквадратическая ошибка по всему изображению.

Краткая характеристика тестовых изображений

Изображение Saturn представляет собой астрофотографию. На черном звездном фоне виден диск планеты и колец. Изображение имеет мало мелких деталей, но достаточно четкие границы. Основное направление деталей изображения – диагональное.

Изображение Parper представляет собой натюрморт из овощей. Присутствуют крупные объекты с высокой контрастностью. Мелких деталей мало.

Изображение Rout – фотография ребенка. Основная часть изображения, несущая максимальную информационную нагрузку – лицо. Изображение малоконтрастно, но в верхней части присутствуют яркие элементы (участки дневного неба через отверстия в заборе).

Изображение *Trees* – искусственное изображение типа гравюры. Изображение имеет неестественно высокую контрастность и обилие мелких деталей. Кроме того, присутствует большое количество больших перепадов яркости. При этом перепады имеют разнообразные направления.

Изображение *Quarter* представляет из себя круглое изображение монеты на белом фоне. Внутри монеты имеются мелкие малоконтрастные детали.

Изображение *Circuit* представляет собой изображение системы технического зрения – фотография подложки микросхемы. Основная особенность – на изображении в основном присутствуют элементы в виде полос горизонтального и вертикального направлений. Область фона присутствует, но фон неравномерный.

Ход работы

1. Под руководством преподавателя или лаборанта включите персональный компьютер лабораторной установки и запустите пакет Matlab. Никакое другое оборудование, входящее в состав лабораторной установки, не включать и не выключать. Окно запущенной программы Matlab должно выглядеть так, как показано на рисунке 26.

2. Создать шаблон отчета по лабораторной работе в текстовом процессоре Word.

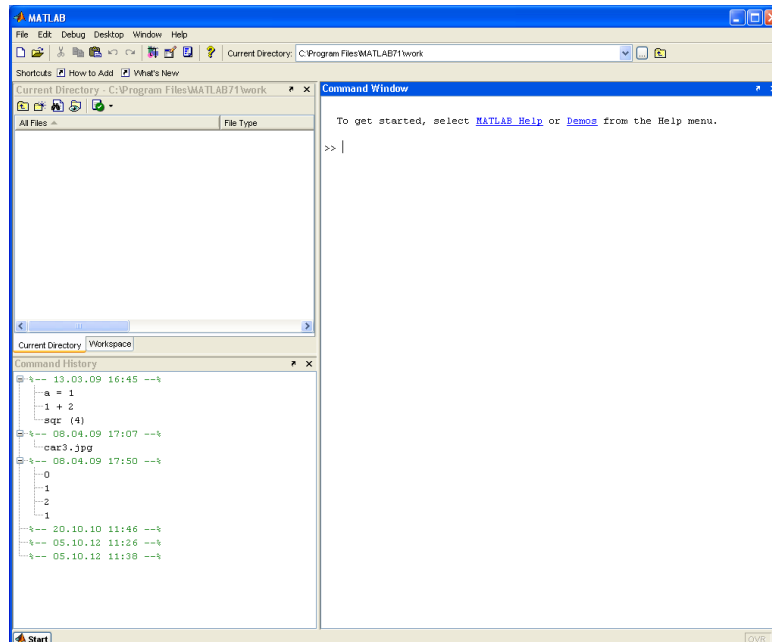


Рисунок 26 – Основное окно программы Matlab

3. В меню «File» выбрать строчку *dctdemo.m*. Должно появиться окно с текстом программы примера *dctdemo.m*. (рис. 27).

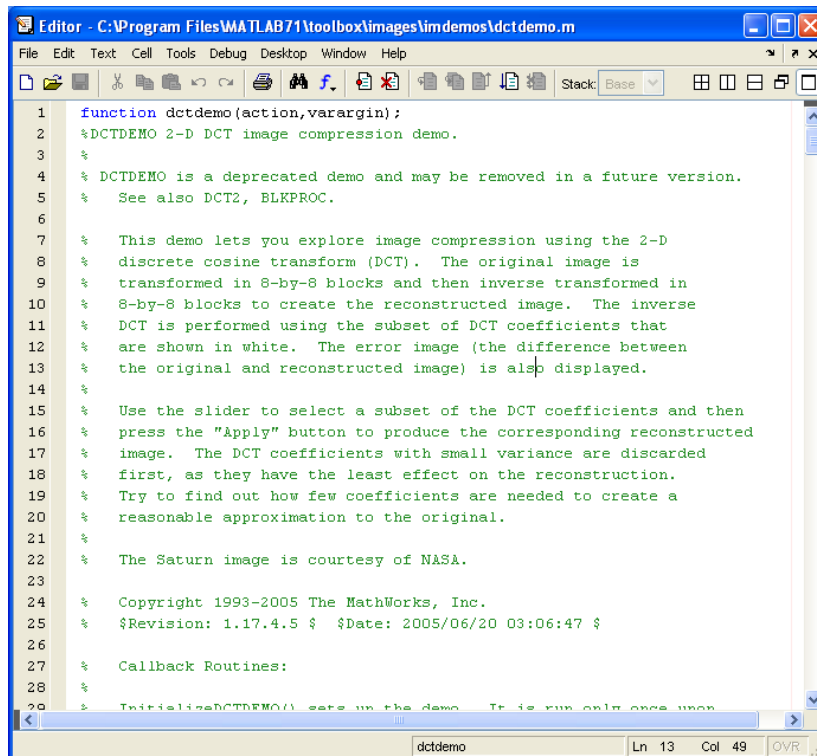


Рисунок 27 – Вид окна программы Matlab с выбранным примером dctdemo

4. В меню примера «Debug» выбрать команду «Run» и запустить пример. В результате появится окно примера (рис. 28).

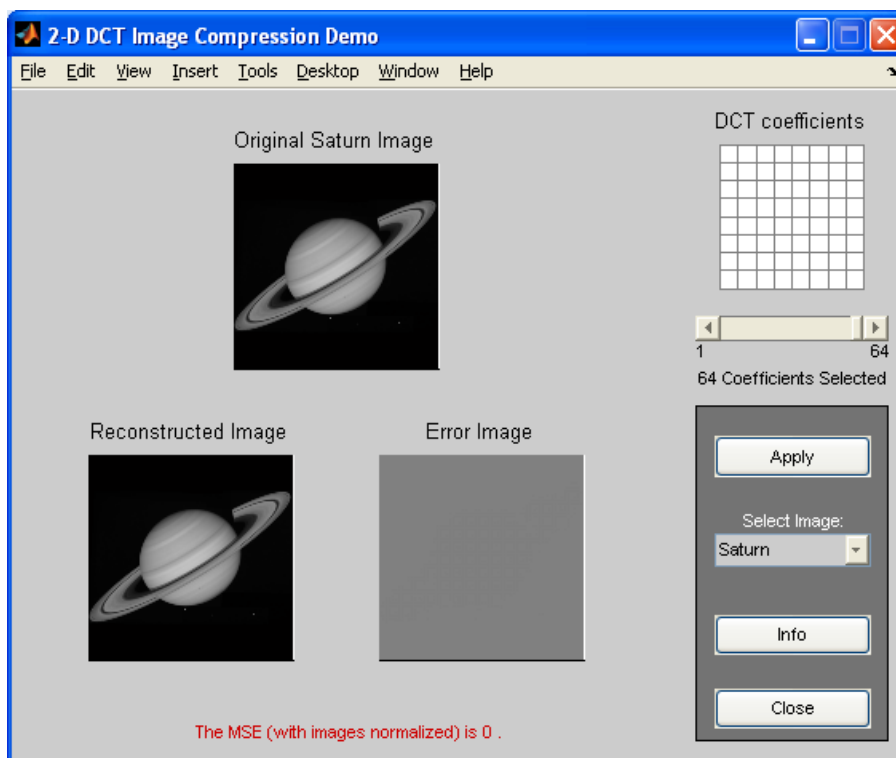


Рисунок 28 – Окно запущенного примера Dctdemo.m

5. Изучить основные элементы примера. Проверить возможность выбора изображений, выбора количества используемых коэффициентов ДКП, наличие исходного, восстановленного изображений и изображения межкадровой разности (ошибок).

6. Установить *изображение Saturn*.

7. С помощью движка установить 1 коэффициент ДКП и нажать кнопку «Apply». Внимательно рассмотреть восстановленное изображение и межкадровую разность. Создать скриншот изображения и занести его в шаблон отчета.

8. Занести значение среднеквадратической ошибки в таблицу 4 (в шаблоне отчета).

Таблица 4

Изображение/ количество коэффициентов ДКП	Saturn	Pepper	Pout	Trees	Quarter	Circuit
1						
2						
3						
4						
8						
12						
16						
20						
24						
30						
40						
50						
64						

9. Установить с помощью движка количество коэффициентов 2 и нажать кнопку «Apply». Внимательно рассмотреть восстановленное изображение и межкадровую разность. Отметить характер изменений в отчете. При необходимости вернуться к изображению с одним коэффициентом. Сделать скриншот изображения и внести в отчет.

10. Установить с помощью движка количество коэффициентов 3 и нажать кнопку «Apply». Внимательно рассмотреть восстановленное изображение и межкадровую разность. Отметить характер изменений в отчете.

11. Выполнить п. 10 с другими значениями числа коэффициентов ДКП в соответствии с таблицей 4. При этом следует отметить в таблице (курсивом) количество коэффициентов ДКП, при котором изображение субъективно оценивается как хорошее, т.е. отсутствуют заметные искажения.

12. Выполнить последовательно п.п. 7 – 11 для оставшихся пяти изображений и полностью заполнить таблицу.

В процессе работы студентам следует сделать 5-6 скриншотов, которые должны подчеркивать наиболее характерные искажения изображений.

Оформление отчета

При оформлении отчета следует пользоваться созданным шаблоном отчета. В шаблон следует занести номер группы и ФИО студентов, участвующих в работе.

В отчете должна быть полностью заполнена таблица 4 и присутствовать скриншоты.

В отчете в свободной форме следует описать характер искажений восстановленного изображения при изменении количества коэффициентов ДКП для каждого из шести тестовых изображений. Указать, при каком количестве коэффициентов ДКП качество восстановленных изображений можно оценить как хорошее.

Вопросы по работе

1. Что такое цифровое телевидение?
2. Что означает аббревиатура JPEG?
3. Какие основные особенности зрения человека используются при сжатии видеосигнала?
4. Какой размер матриц обычно используется в ДКП и почему?
5. Что такое коэффициенты дискретного косинусного преобразования?
6. Каков физический смысл базовых функций ДКП?
7. Почему квантование коэффициентов ДКП создает менее заметные искажения, чем квантование самого изображения?

3 Лабораторная работа «Изучение методов построения панорам»

Цель работы

Изучить основные элементы метода построения панорам на основе сшивки изображений, полученных с помощью цифровой камеры.

Введение и основные теоретические сведения

Под панорамой обычно понимают изображение, у которого поле зрения по одной из координат (чаще всего по горизонтали) существенно превышает поле зрения по второй координате. Предельным случаем является т.н. сферическая панорама, в основе которой лежит собранное из множества отдельных кадров изображение в сферической (эквидистантная, equirectangular, sphere) или кубической проекции. Характерной чертой сферических панорам является максимально возможный угол обзора пространства (360×180 градусов).

В связи с тем, что сферическая проекция вносит специфические искажения в изображение (особенно в верхней и нижней части), сферические панорамы практически никогда не демонстрируются в печатном виде или в виде обычного графического файла [13, 14].

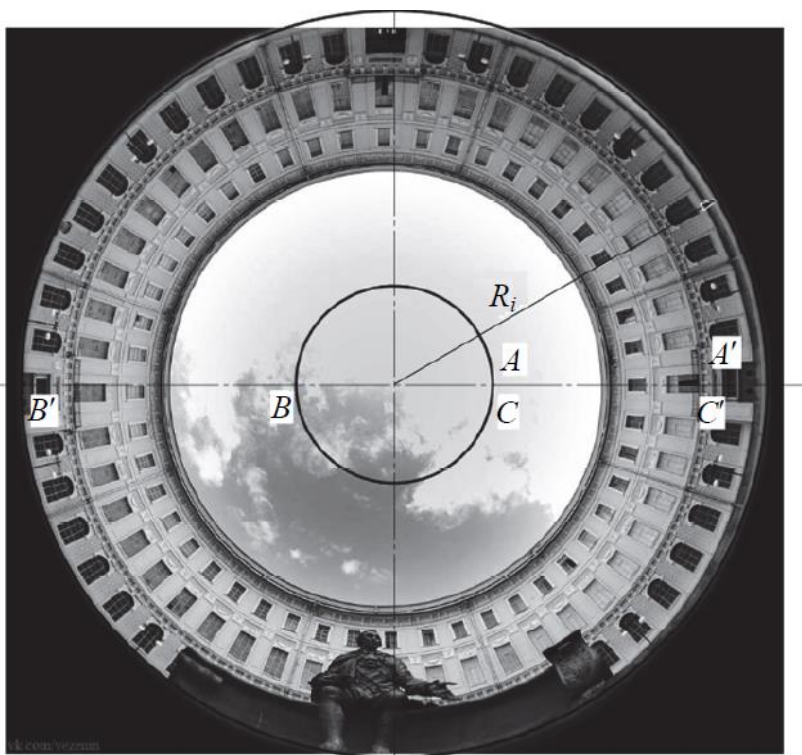


Рисунок 29 – Входное изображение, полученное с помощью камеры с объективом «Рыбий глаз»

Основным способом демонстрации является компьютерная визуализация с использованием специального программного обеспечения. У зрителя создаётся иллюзия присутствия внутри сферы, на внутреннюю

поверхность которой «натянута» изображение окружающего пространства. При этом оптические искажения (сферические аберрации) не видны (рис. 29, 30). К тому же, как правило, сферические панорамы наделяются инструментами управления просмотром, позволяющими изменять направление просмотра (вверх-вниз, вправо-влево), а также приближать или отдалять изображение. Благодаря всему этому зритель видит место, где производилась съёмка, так, как если бы находился там сам.



Рисунок 30 – Пример сферической панорамы после использования одного из методов компенсации сферической аберрации

Основным источником изображений для синтеза панорам является цифровая камера.

Возможны два основных способа получения изображений.

Первый способ характеризуется использованием только одного кадра, формируемого камерой с объективом с полем зрения около 180 градусом (т.н. «рыбий глаз») или специальным зеркальным кольцевым объективом (рис. 31).



Рисунок 31 – Сферический объектив «Рыбий глаз» (справа) и кольцевой зеркальный объектив (слева)

В этом случае для формирования панорамы сшивки кадров не требуется. Полученный кадр имеет в своем составе все элементы

изображения. Требуется только коррекция аберраций объектива. Основным недостатком такого метода является получение панорамы сравнительно небольшого разрешения, соответствующему формату используемого матричного фотоприемника в камере. Так как при просмотре панорам обычно используется оконный режим с небольшим рабочим полем зрения (фактически дополнительное увеличение изображения), то разрешение внутри такого окна, особенно с учетом работы алгоритма коррекции аберраций, будет существенно меньше, чем исходное разрешение используемого матричного фотоприемника. Тем не менее, такой метод построения панорам получил большое распространение благодаря простому и быстрому процессу съемки. Чаще всего именно такой метод используется при формировании панорам в сервисах Google и Yandex (рис. 32).

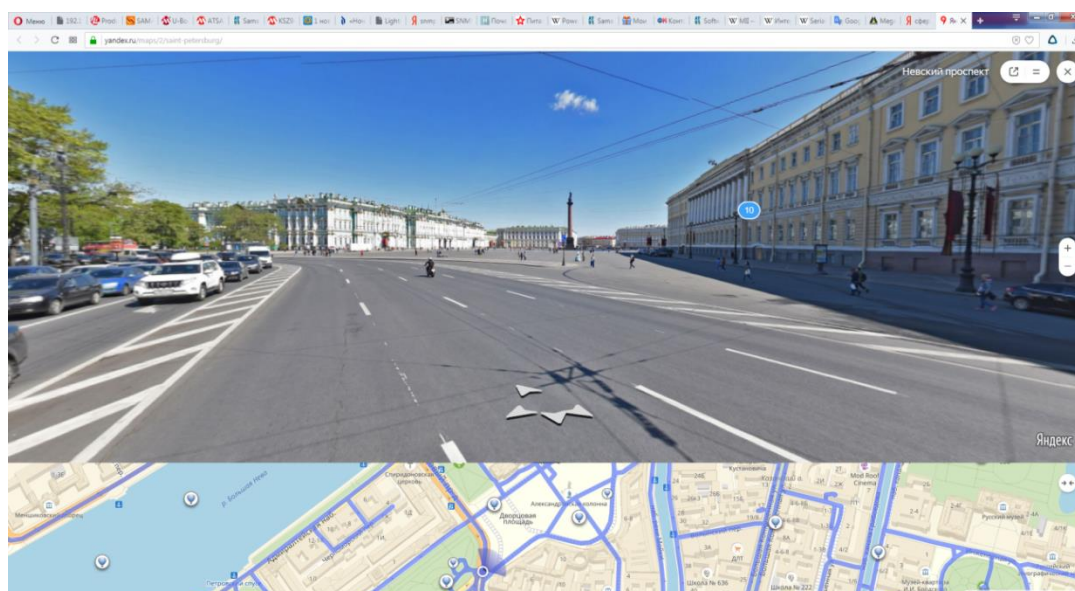


Рисунок 32 – Фрагмент панорамы Санкт-Петербурга на сервисе Yandex карты

Второй способ предполагает использование серии изображений, полученных с помощью камеры со сравнительно небольшим полем зрения. Чаще всего при съемке этих изображений камера передвигается по одной из осей (обычно по горизонтали) с использованием специального оборудования или, в крайнем случае, вручную (рис. 33). Существует вариант использования сразу нескольких камер, положение которых зафиксировано друг относительно друга.



Рисунок 33 – Специальное поворотное устройство (слева) и комплекс из четырех камер (справа) для съемки сферических панорам

В этом случае отдельные кадры с малым полем зрения требуется объединить в одно изображение с большим полем зрения вплоть до полусферы или сферы (рис. 34).



Рисунок 34 – Два кадра, полученные с камеры (вверху) и результат их сшивки (внизу)

При этом требуется использование технологии совмещения изображений с использованием одинаковых или похожих элементов в соседних изображениях. Обычно эта технология называется сшивкой изображений. Процесс сшивки требует наличия некоторых характерных хорошо различимых элементов небольшого размера в паре сшиваемых изображений. Определяются координаты этих элементов в обоих изображениях, и после этого, опираясь на эти элементы и их координаты, происходит совмещение подобных областей в изображениях, что является необходимым условием успешной сшивки. Такую сшивку требуется выполнить для всех изображений, входящих в будущую панораму.

Наиболее известным методом определения опорных точек является метод SURF. Этот метод решает две задачи – поиск особых точек изображения и создание их дескрипторов, инвариантных к масштабу и вращению. Это значит, что описание ключевой точки будет одинаково, даже если образец изменит размер и будет повернут (здесь и далее мы будем говорить только о вращении в плоскости изображения). Кроме того, сам поиск ключевых точек тоже должен обладать инвариантностью, так, чтобы повернутый объект сцены имел тот же набор ключевых точек, что и образец.

Метод ищет особые точки с помощью матрицы Гессе, детерминант матрицы Гессе достигает экстремума в точках максимального изменения градиента яркости. Он хорошо детектирует пятна, углы и края линий.



Рисунок 35 – Показаны особые точки изображения в образце (слева) и на сцене (справа)

Несмотря на то, что сцена имеет другой масштаб, угол обзора и частично заслонена другим объектом, ключевые точки достаточно точно идентифицируются [14].

Гессиан инвариантен относительно вращения, но не инвариантен масштабу. Поэтому SURF использует разномасштабные фильтры для нахождения гессианов.

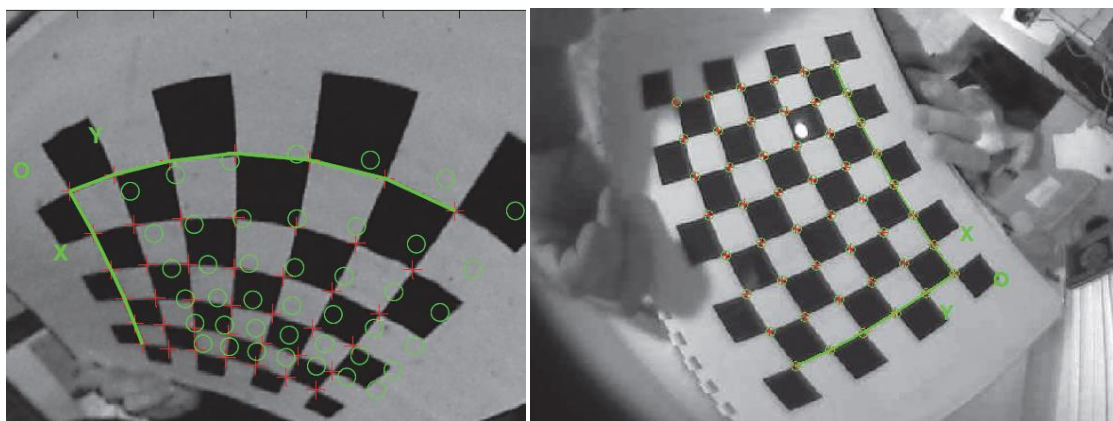
Для каждой ключевой точки считается направление максимального изменения яркости (градиент) и масштаб, взятый из scale-коэффициента матрицы Гессе. Градиент в точке вычисляется с помощью фильтров Хаара.

После нахождения ключевых точек SURF формирует их дескрипторы. Дескриптор представляет собой набор из 64 (либо 128) чисел для каждой ключевой точки. Эти числа отображают направление градиента вокруг ключевой точки. Поскольку ключевая точка представляет собой максимум гессиана, то это гарантирует, что в окрестности точки должны быть участки с разными градиентами. Таким образом, обеспечивается дисперсия (различие) дескрипторов для разных ключевых точек [11, 14].

Направление градиента окрестностей ключевой точки считается относительно направления градиента вокруг точки в целом (но всей окрестности ключевой точки). Таким образом, достигается инвариантность дескриптора относительно вращения. Размер же области, на которой считается дескриптор, определяется масштабом матрицы Гессе, что обеспечивает инвариантность относительно масштаба. Направление градиента также считается с помощью фильтра Хаара. [13]

Основной проблемой является относительно трудоемкий процесс получения первичных изображений, а также необходимость наличия опорных элементов в изображении. Кроме того, при определении координат следует учитывать искажения вследствие сферической аберрации. При слишком большом поле зрения исходных изображений на краях поля зрения могут возникнуть такие искажения, которые сделают невозможным определение координат опорных элементов в соседних изображениях с заданной точностью. Для коррекции таких искажений требуется предварительная калибровка камеры с использованием тест-объекта.

На рисунке 36 красные кресты – найденные калибровочные точки тест-объекта, зеленые кружки – результат проецирования рассчитанных в процессе калибровки трехмерных координат калибровочных точек обратно на изображение: пример неверного определения калибруемых параметров (рис. 36 а); результат экспериментальной калибровки – положение калибровочных точек и перепроецированных точек совпадает, что подтверждает верное определение калибруемых параметров оптико-электронной системы (рис. 36 б); размер каждого квадрата паттерна – 50 мм [14, 16].



а

б

Рисунок 36 – Результаты калибровки с помощью инструментария «OCamCalib Toolbox»

Главным достоинством полученных таким способом панорам является высокое разрешение составного изображения, которое позволяет просматривать элементы панорамы в небольшом угле поля зрения фактически с исходным разрешением матричного фотоприемника камеры. Несмотря на трудоемкость, такой метод получил широкое распространение при реализации функции синтеза панорам в цифровых фотоаппаратах и смартфонах. Кроме того полученные таким образом панорамы, используются в сервисах демонстрации высококачественных панорам объектов (рис. 37) [15]. Аналогичные панорамы с разрешением до 320 гигапикселей представлены на сайте <https://www.360cities.net>.

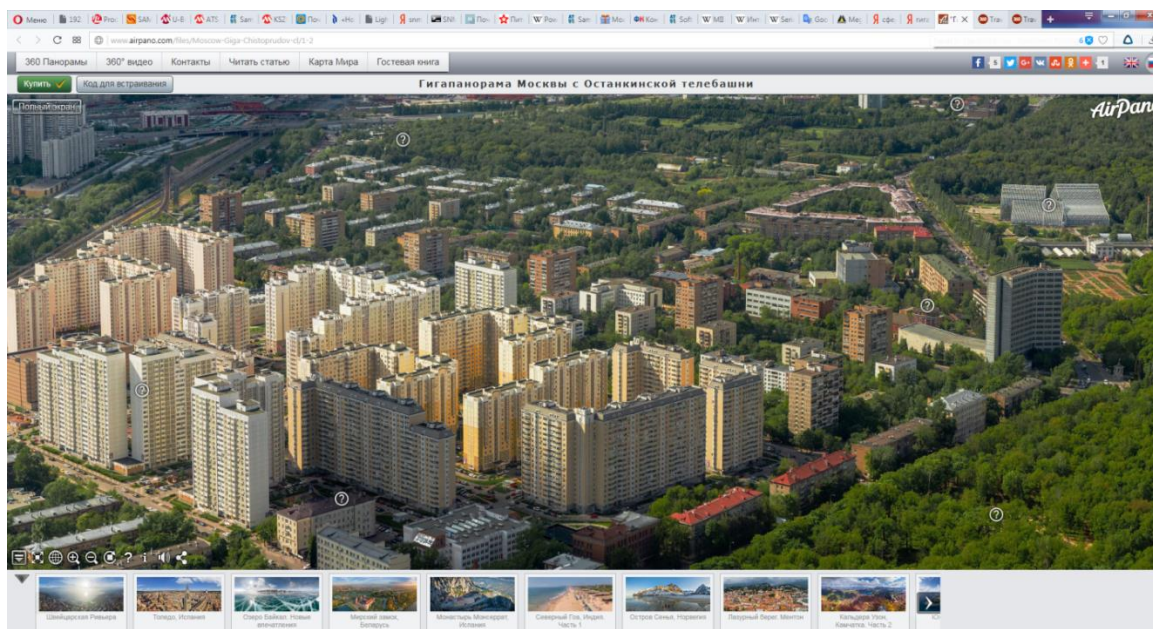


Рисунок 37 – Пример высококачественной панорамы Москвы, полученной с Останкинской телебашни

Состав лабораторной установки

Универсальная лабораторная установка, включающая в себя:
Персональный компьютер и монитор;
Камера VEC-545 с интерфейсом USB с объективом-трансфокатором на жесткой поворотной платформе;
Тестовое изображение на подставке;
Светодиодный осветитель;
Блок питания осветителя.
Программное обеспечение включает в себя:
Операционную систему Windows 7 или старше;
Драйверы камер и других устройств;
Пакет Matlab.

Ход работы

С помощью преподавателя или лаборанта включить лабораторную установку. Самостоятельно ничего не включать!

1. Запустить пакет Matlab, щелкнув по ярлыку на рабочем столе (рис. 38).

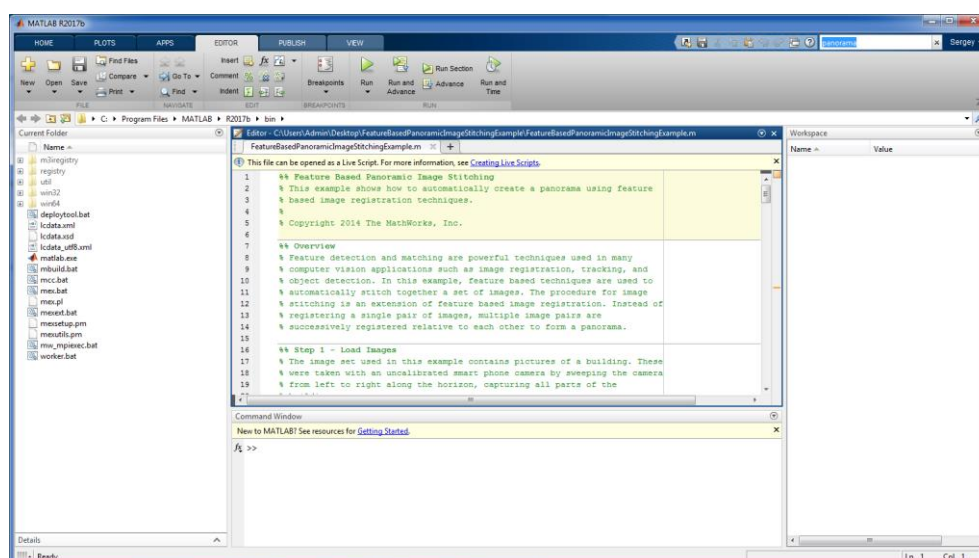


Рисунок 38 – Вид рабочего окна пакета Matlab после открытия

2. Загрузить файл FeatureBasedPanoramicImageStitchingExample.m. Как правило, при открытии программы Matlab он уже загружен. В противном случае воспользоваться поиском (см. следующий пункт) и загрузить файл кнопкой Open Script.

3. Найти в поиске по ключевому слову Panorama описание **Feature Based Panoramic Image Stitching**. (рис. 39).

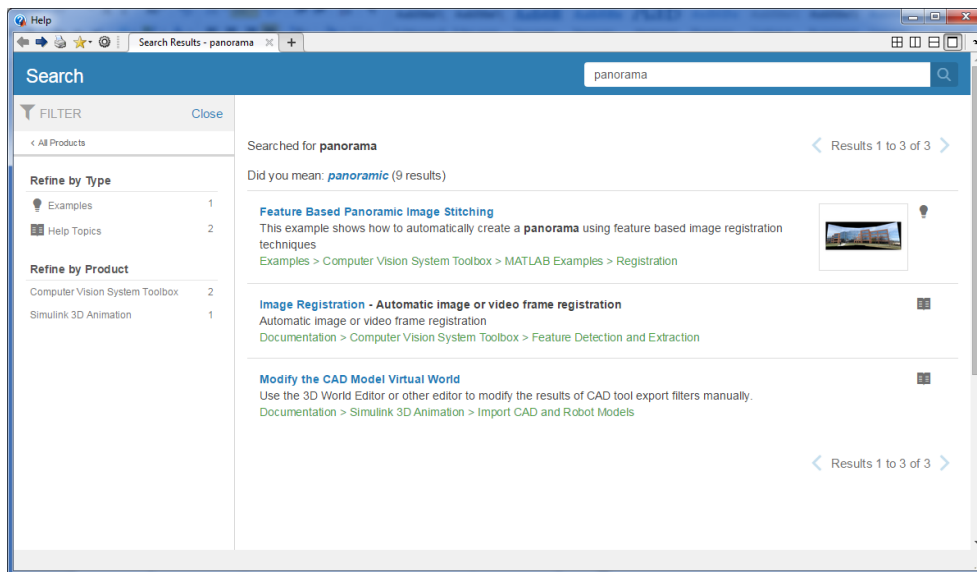


Рисунок 39 – Результат поиска по ключевому слову Panorama

4. Открыть описание демонстрационной программы FeatureBasedPanoramicImageStitchingExample.m (рис. 40).

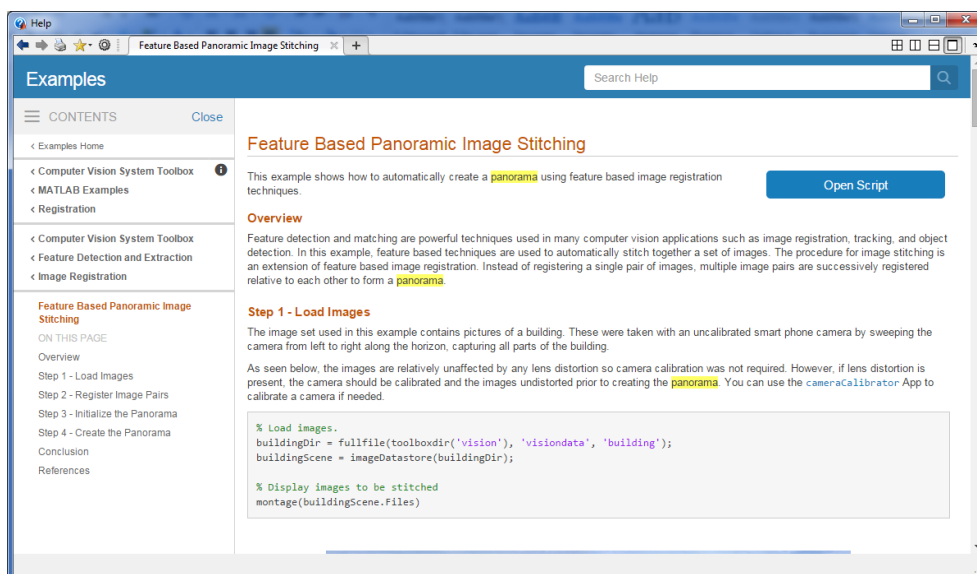


Рисунок 40 – Описание программы FeatureBasedPanoramicImageStitchingExample.m

5. Ознакомиться со структурой программы, особенно обратить внимание на способ получения исходных цифровых изображений для примера с изображениями здания.

6. Далее ознакомиться со способом визуализации исходных изображений, разбиением исходных изображений на пары, фрагментом, описывающим работу алгоритма SURF, получением особых точек, совмещением изображений и, в конечном счете – построением панорамы.

7. Запустить в пакете Matlab приложение FeatureBasedPanoramicImageStitchingExample.m.

8. Дождаться выполнения тестового примера и сделать скриншоты исходных изображений и полученной панорамы (рис. 41).

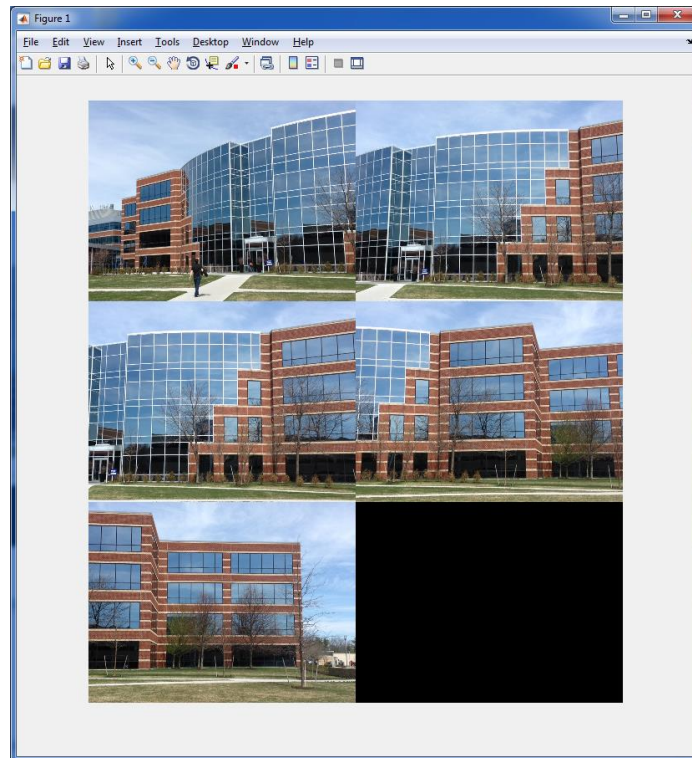


Рисунок 41 – Вид окна с исходными изображениями

9. Найти в исходном тексте программы фрагмент задания пути к папке с исходными изображениями. Заменить путь Building на Building1. Запустить программу вновь и дождаться завершения работы программы. Сохранить полученные окна с исходными изображениями помещения лаборатории и с изображением панорамы (рис. 42-44). Сохранить полученные скриншоты.

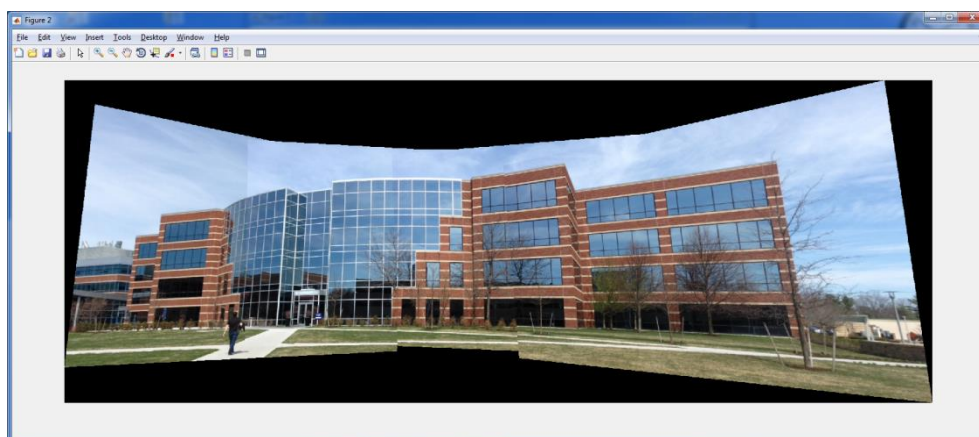


Рисунок 42 – Вид окна с результатом построения панорамы

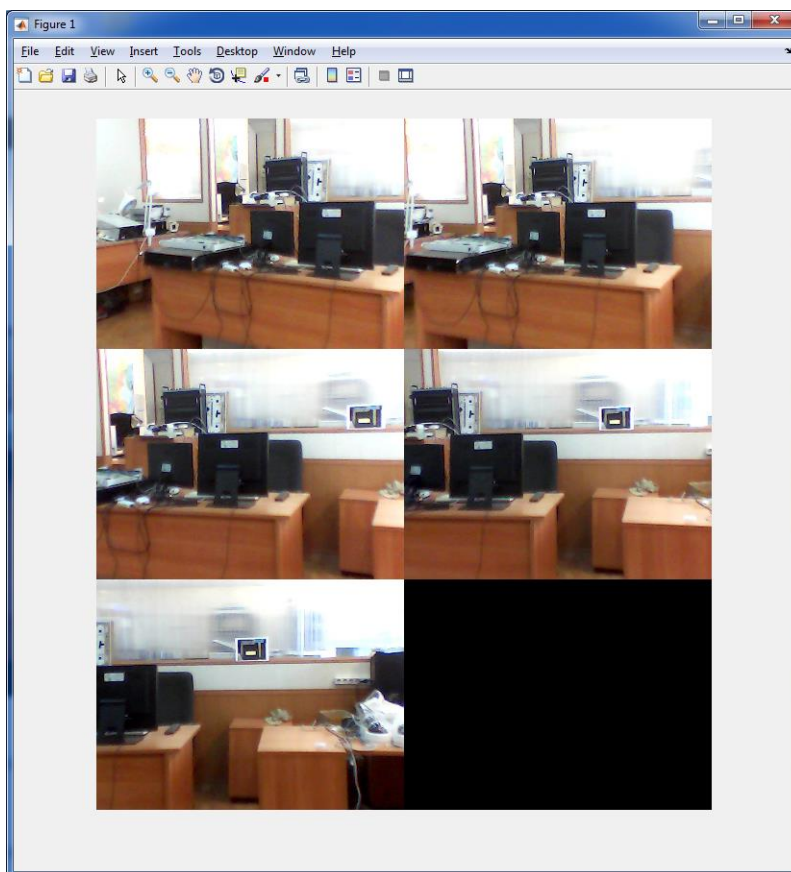


Рисунок 43 – Исходные изображения лаборатории

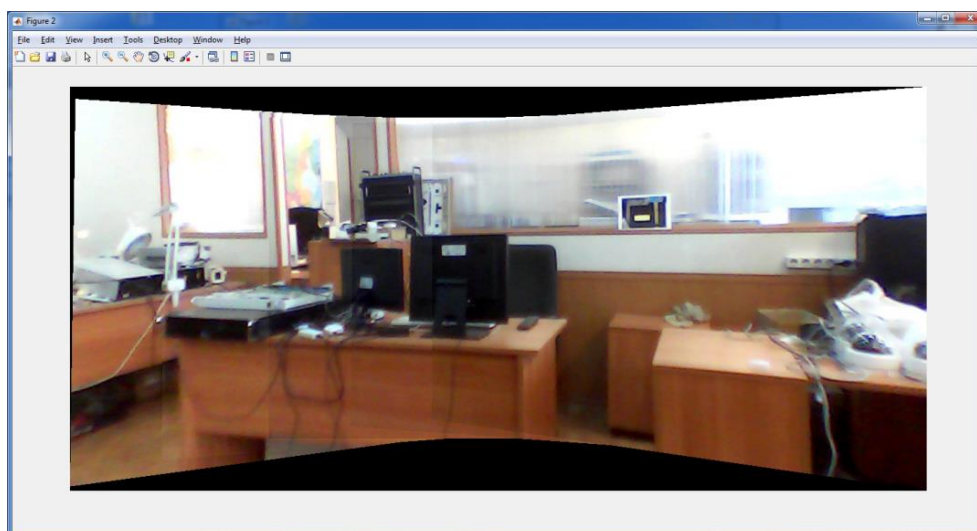


Рисунок 44 – Панорама лаборатории

10. Сменить путь на Building9 и запустить приложение. Дождаться получения результатов. Сохранить полученные скриншоты исходных изображений днища модели легкового автомобиля и неудачный результат синтеза панорамы (рис. 45).

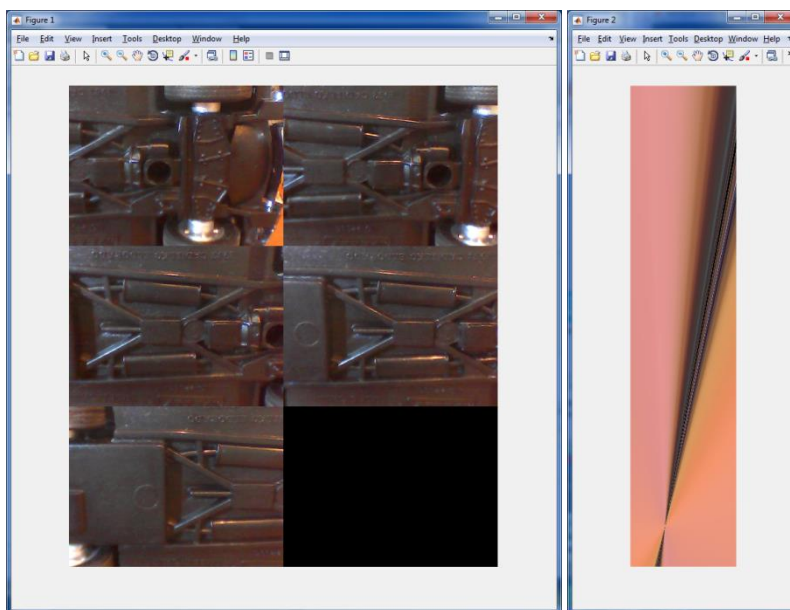


Рисунок 45 – Изображений днища модели легкового автомобиля и неудачный результат синтеза панорамы

11. Временно свернуть все окна программы Matlab и запустить на рабочем столе программу обслуживания цифровой камеры Webcam videocap. Откроется основное окно программы (рис. 46).

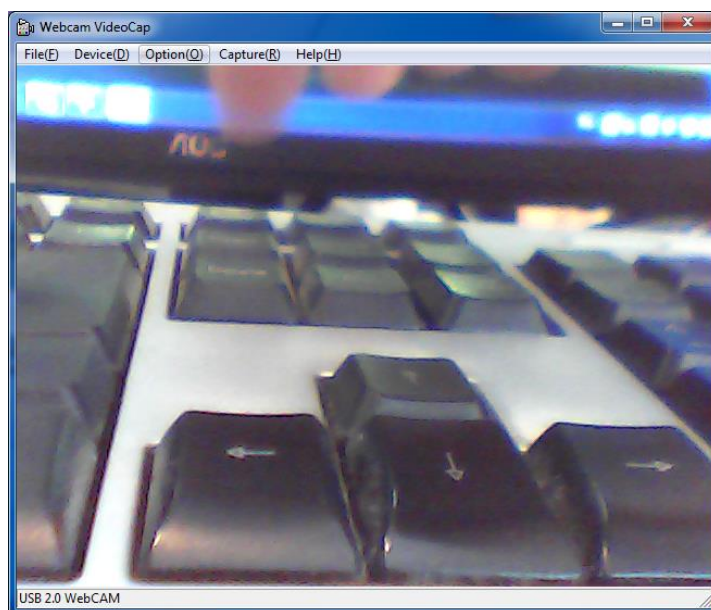


Рисунок 46 – Основное окно программы Webcam videocap

12. Убедиться в качественном изображении с камеры и при необходимости настроить его с помощью преподавателя или лаборанта, используя вкладку Option. Если есть регулировка угла поля зрения, то установить наименьший угол (наибольшее фокусное расстояние объектива).

13. Выполнить съемку панорамы тест-объекта с использованием пяти исходных кадров. Для этого следует установить камеру на выбранную часть объекта и во вкладке Capture выбрать пункт Snap Shot. Изображения следует выполнить с перекрытием 40 – 50% (рис. 47).



Рисунок 47 – Получение исходных изображений с перекрытием 40-50%

14. Изображение сохранится в папке, которую можно открыть, используя команду во вкладке File - Open Capture File Folder (рис. 48).

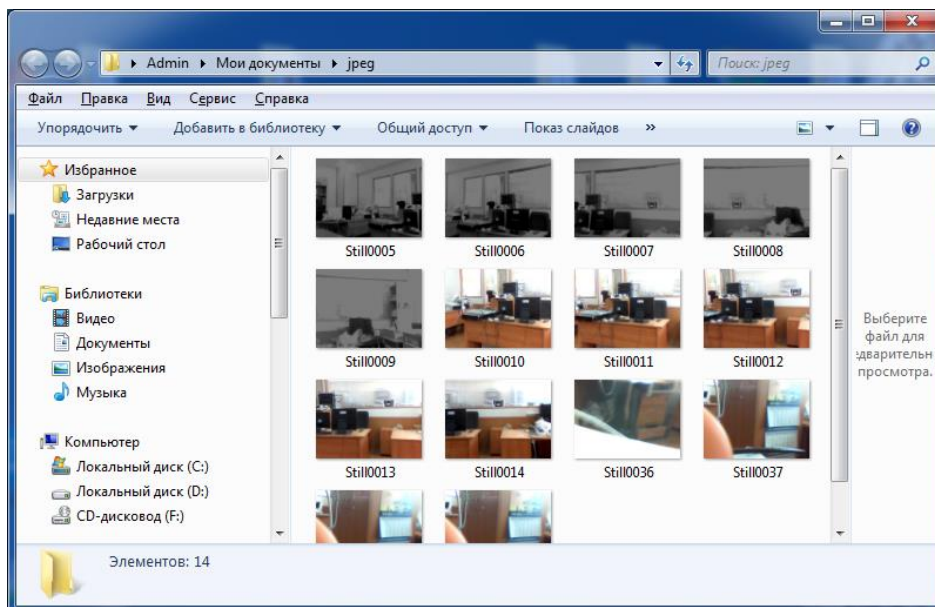


Рисунок 48 – Содержимое папки с сохраненными изображениями

15. Найдите папку Matlab с файлами, содержащими примеры Building. Не удаляя существующие папки и файлы, создать рядом папку Building20 и скопировать или перенести туда полученные вами изображения с камеры. В тексте программы установите путь Building20 и запустите программу. Дождитесь окончания ее выполнения и сохраните скриншоты.

16. Выполните п. 13-15 с перекрытием 15-20 % (рис. 49).



Рисунок 49 – Получение исходных изображений с перекрытием 15-20%

17. Если есть регулировка угла поля зрения, то установите наибольший угол (наименьшее фокусное расстояние объектива) и выполните дважды п.п. 13 – 16 для этого поля зрения (соответственно, с перекрытием 40-50% и с перекрытием 15-20%). Сохраните полученные скриншоты.

18. Для подготовки отчета сохраните все результаты работы (скриншоты и исходный текст программы).

19. С помощью преподавателя или лаборанта выключите лабораторную установку.

Оформление отчета

Отчет должен быть оформлен в электронном виде и распечатан. За основу можно взять прилагаемый шаблон Ir.dot. В имеющемся шаблоне следует заполнить обязательные поля, включающие ФИО и группу студента, дату выполнения работы, полученные результаты, а также основные скриншоты с полученными изображениями. В приложении привести текст исходной программы. При необходимости вставленные в шаблон рисунки следует отмасштабировать.

Вопросы по работе

1. Что такое панорама и как она используется?
2. Какие методы используются для построения панорам.
3. Назовите основные элементы демонстрационной программы для построения панорам.
4. Что такое сшивка изображений?
5. Назовите основные особенности метода SURF.
6. Какие параметры и особенности изображения влияют на качество построения панорам?
7. Почему в ряде случаев панорама не формируется?

4 Лабораторная работа «Исследование алгоритмов улучшения субъективного восприятия изображений»

Цель работы

Изучить основные методы цифровой обработки видеосигналов с целью улучшения субъективного восприятия изображений с телевизионных камер.

Краткие теоретические сведения

В условиях, когда телевизионная камера установлена неподвижно, а объект движется может возникнуть эффект смаза изображения. Это связано с тем, что за время накопления камеры объект проходит некоторое расстояние. Если время накопления относительно велико, то смаз становится не только заметен, но и существенно влияет на качество изображения.

Процесс смаза может быть описан операцией свертки. Обратный свертке процесс называется деконволюцией.

Для того, чтобы метод деконволюции наилучшим образом восстанавливал изображение, необходимо как можно точнее знать параметры смаза изображения.

Ниже приводится пример работы метода деконволюции для восстановления изображений вагонов (рис. 50-52).

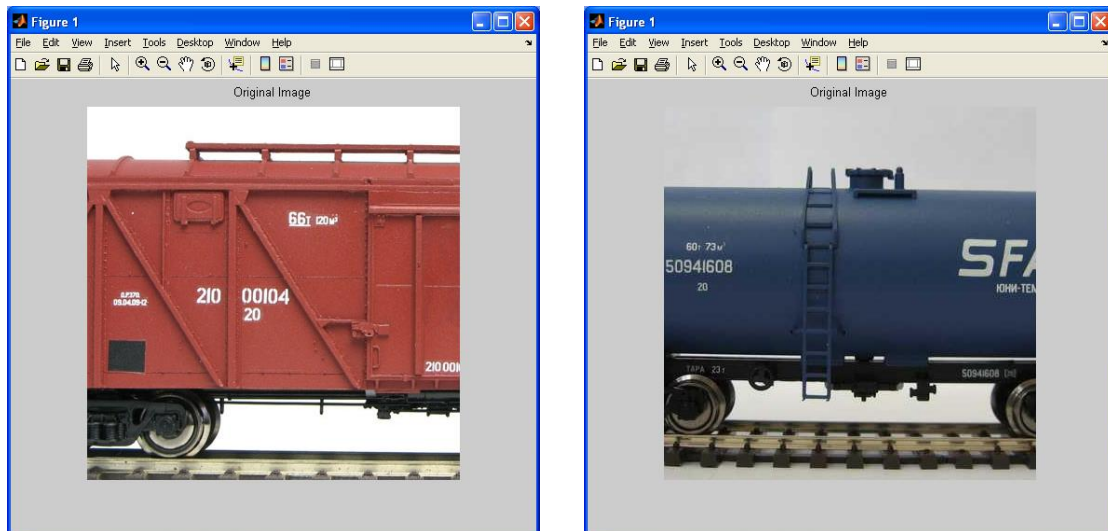


Рисунок 50 – Пример стоящих на месте грузового вагона и цистерны. Смаз отсутствует

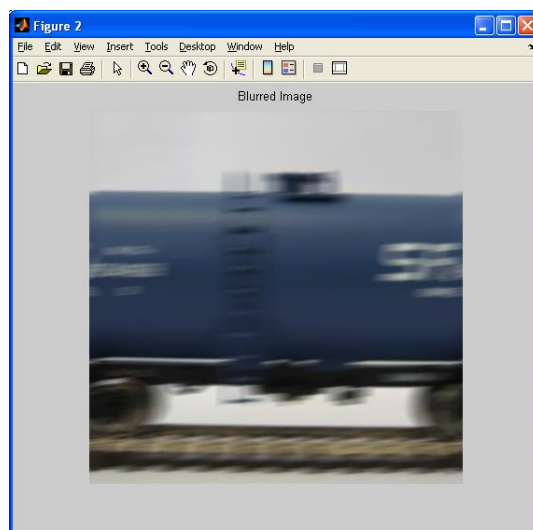
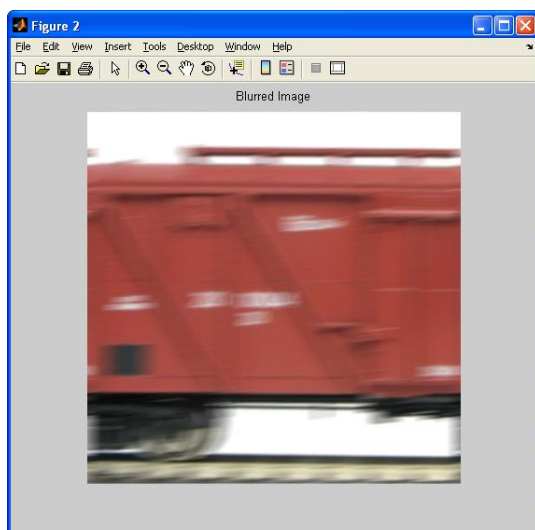


Рисунок 51 – Пример движущихся грузового вагона и цистерны. Смаз очень заметен

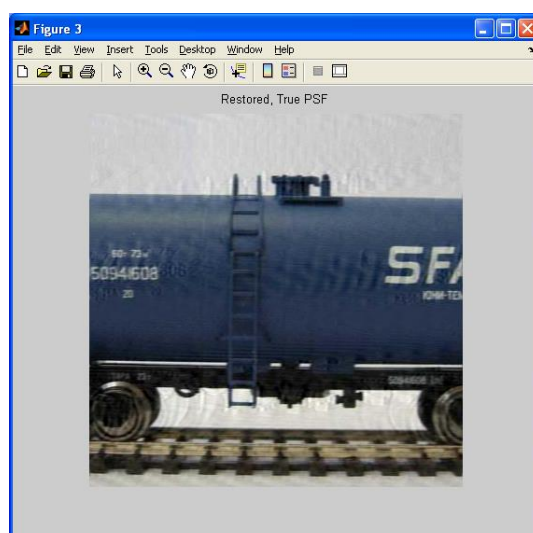
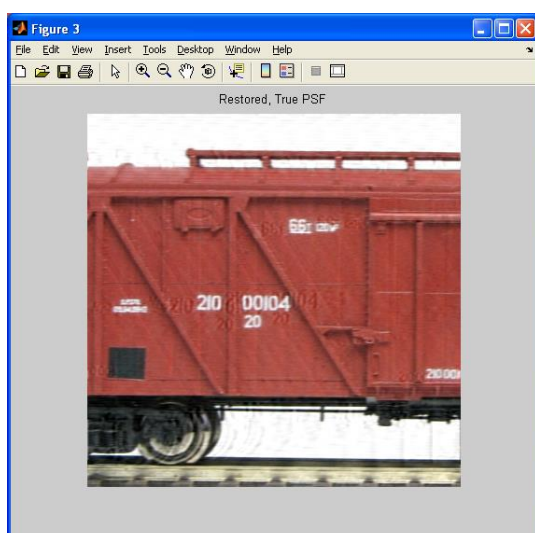


Рисунок 52 – Пример восстановления изображений методом деконволюции при известной величине смаза

Результат работы метода деконволюции сильно зависит от исходных данных изображения. В частности, довольно большое значение имеет блочная структура изображения (рис. 53), которое используют методы на основе дискретного косинусного преобразования. Если компрессия изображения выше определенного уровня, при котором начинает проявляться блочная структура, то результат деконволюции может быть недостаточно эффективным.

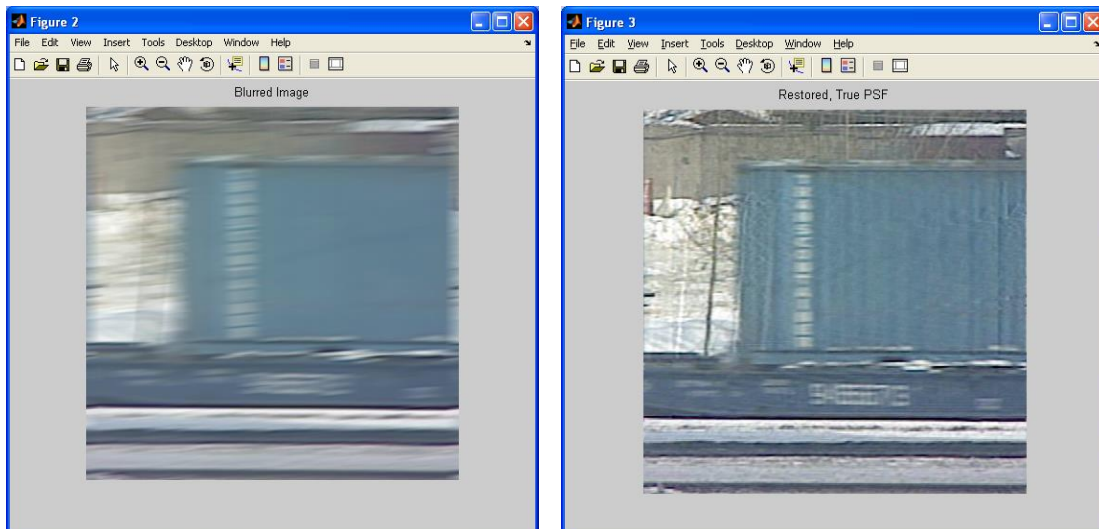


Рисунок 53 – Пример восстановления изображения при наличии блочного эффекта

Изучение метода деконволюции производится с помощью программы Matlab.

Ход работы

При подготовке отчета следует воспользоваться файлом с шаблоном отчета.

Запустите программу Matlab, щелкнув мышью по ярлыку Matlab 7.1. Появится окно программы Matlab.

Проверим наличие рабочих файлов, которые потребуются при выполнении работы.

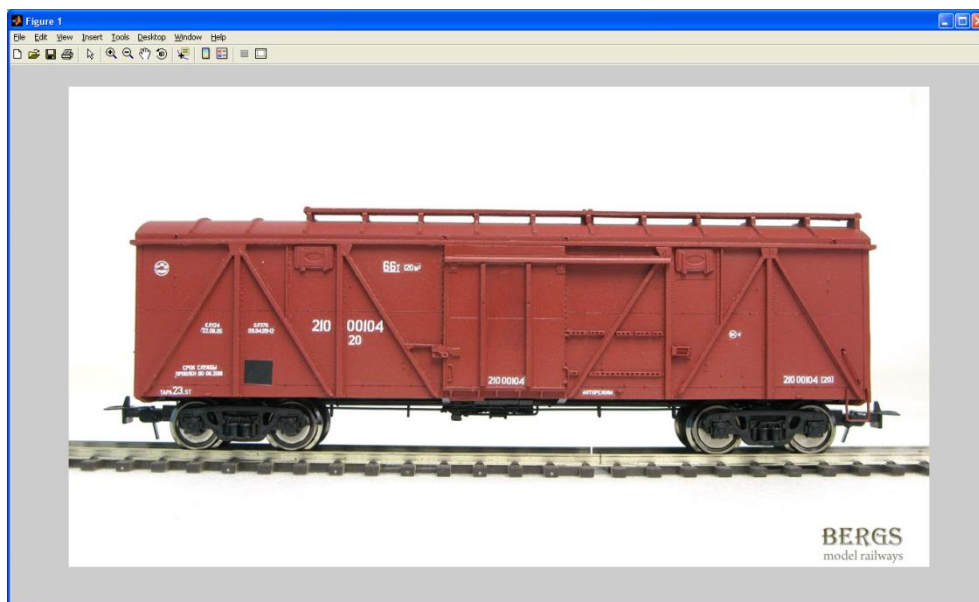


Рисунок 54 – Пример файла с изображением

Потребуется три изображения poluvag.jpg, cist.jpg auto.jpg.

Проверим их наличие прямо в рабочей области Matlab. Для этого следует набрать команду

```
>> I = imread('poluvag.jpg');
```

```
>>
```

Matlab прочитает файл с изображением в массив I.

Выполним команду

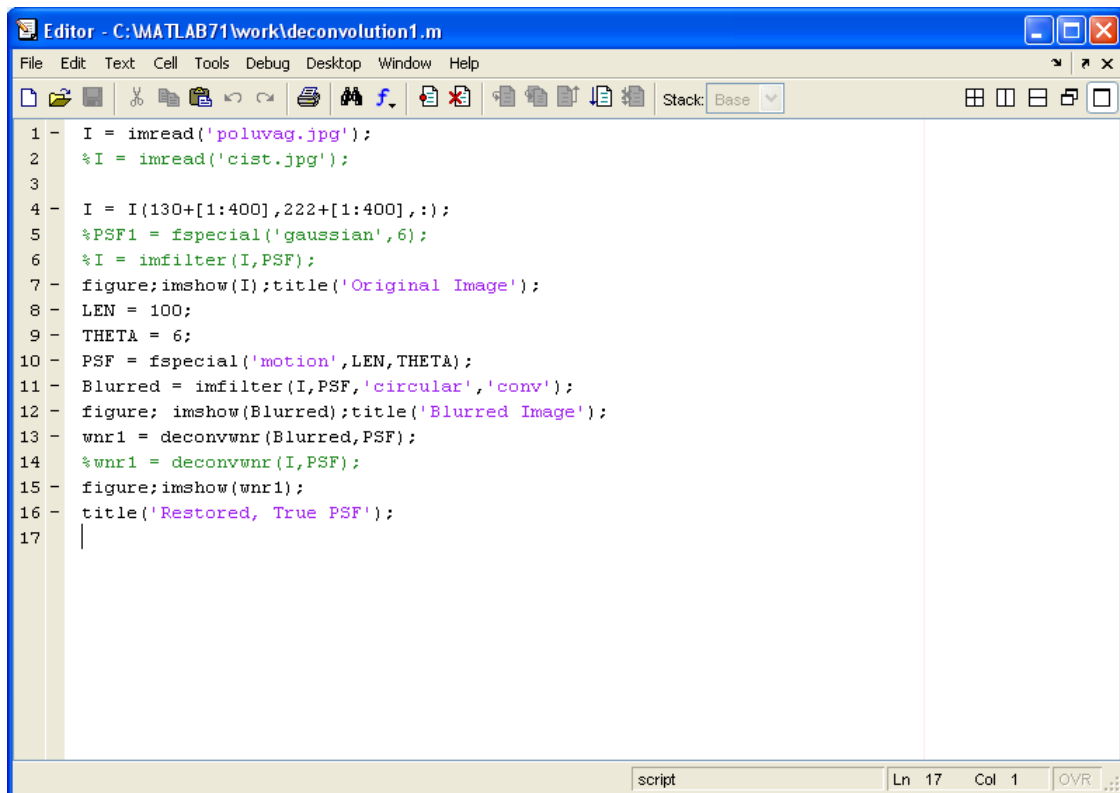
```
>> imshow(I)
```

Должно открыться изображение в отдельном окне (рис. 54).

Закройте это изображение и повторите действия с другими изображениями cist.jpg и 21.jpg.

Все три изображения в виде скриншотов поместите в шаблон отчета.

Откройте файл с программой deconvolution1.m. Должно появиться окно (рис. 55).



```
Editor - C:\MATLAB71\work\deconvolution1.m
File Edit Text Cell Tools Debug Desktop Window Help
I = imread('poluvag.jpg');
%I = imread('cist.jpg');
I = I(130+[1:400],222+[1:400],:);
%PSF1 = fspecial('gaussian',6);
%I = imfilter(I,PSF);
figure;imshow(I);title('Original Image');
LEN = 100;
THETA = 6;
PSF = fspecial('motion',LEN,THETA);
Blurred = imfilter(I,PSF,'circular','conv');
figure; imshow(Blurred);title('Blurred Image');
wnr1 = deconvwnr(Blurred,PSF);
%wnr1 = deconvwnr(I,PSF);
figure;imshow(wnr1);
title('Restored, True PSF');
|
script Ln 17 Col 1 OVR
```

Рисунок 55 – Окно программы deconvolution1

Программа выполняет две задачи:

Моделирование смаза изображения и восстановление смаза методом деконволюции.

В первой строчке программы имеется имя исходного файла.

Восьмая строчка имеет параметр LEN – это смещение изображения в пикселах в горизонтальном направлении, которое приводит к смазу.

Девятая строчка имеет параметр ТНЕТА – аналогичное смещение по вертикали.

Установите в первой строчке название первого файла `roluvag.jpg`. Установите параметр `LEN = 30`. Запустите программу на выполнение выбрав в меню `Debug` пункт `Run`.

После выполнения программы получатся три изображения (они могут быть закрыты друг другом) (рис. 56). Рассмотрите скриншоты смазанного и восстановленного изображений. Сравните их с исходным изображением.

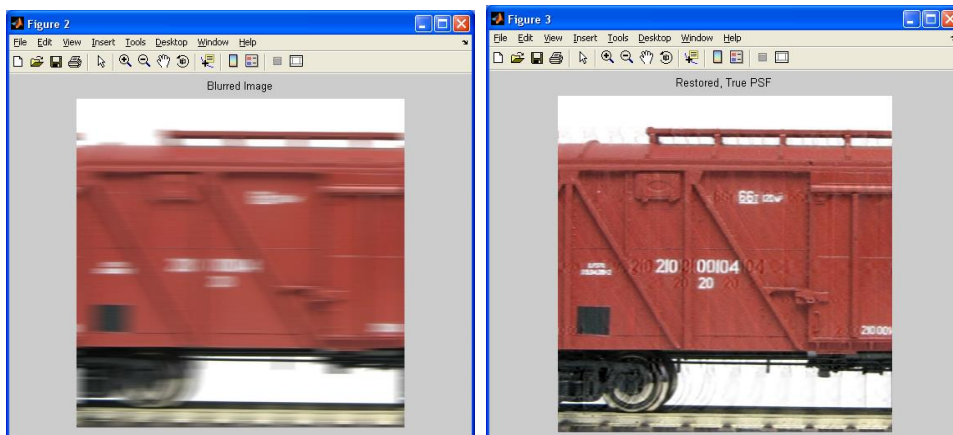


Рисунок 56 – Изображения, полученные в результате выполнения программы `deconvolution1`

Для данного изображения характерным является именно смаз по горизонтали, поэтому параметр `THETA` можно всегда устанавливать `0`.

Закройте все изображения и повторите процесс восстановления при установке следующих значений параметра `LEN`: `2`, `5`, `15`, `40`, `70`, `100`, `150`.

Соберите скриншоты при заданных параметрах смаза.

Введите другой файл изображения, для чего, например, можно установить знак комментария `%` в первой строчке программы и снять знак комментария во второй строчке.

Начните эксперименты со значения `LEN = 30` (рис. 57).

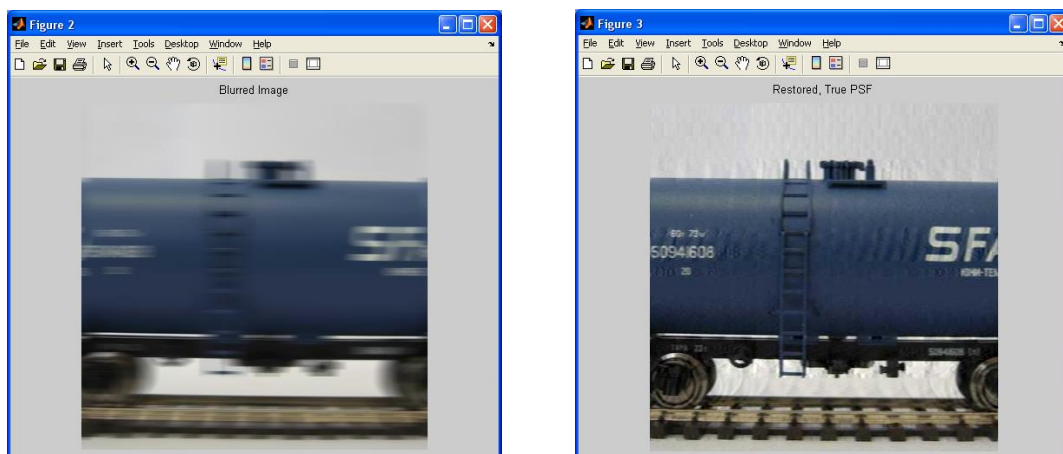


Рисунок 57 – Пример, получаемых изображений в результате выполнения программы `deconvolution1`

Закройте все изображения и повторите процесс восстановления при установке следующих значений параметра LEN: 2, 5, 15, 40, 70, 100, 150.

Соберите скриншоты при заданных параметрах смаза.

Закройте программу deconvolution1.m и откройте программу deconvolution.m.

В отличие от первой части работы, мы имеем дело с реальным смазанным изображением, полученным с камеры с большим временем накопления.

```
Editor - C:\MATLAB71\work\deconvolution.m
File Edit Text Cell Tools Debug Desktop Window Help
1
2 - I = imread('21.jpg');
3   %I = imread('auto.jpg');
4
5 - I = I(130+[1:400],222+[1:400],:);
6
7 - figure; imshow(I);title('Original Image');
8 - LEN = 5;
9 - THETA = 0;
10 - PSF = fspecial('motion',LEN,THETA);
11
12 - wnr1 = deconvwnr(I,PSF);
13 - figure; imshow(wnr1);
14 - title('Restored, True PSF');
15
script Ln 8 Col 8 OVR
```

Рисунок 58 – Окно программы deconvolution.m

Запустите программу, используя пункты меню Debug и Run. Появятся исходное и восстановленное изображения (рис. 59).

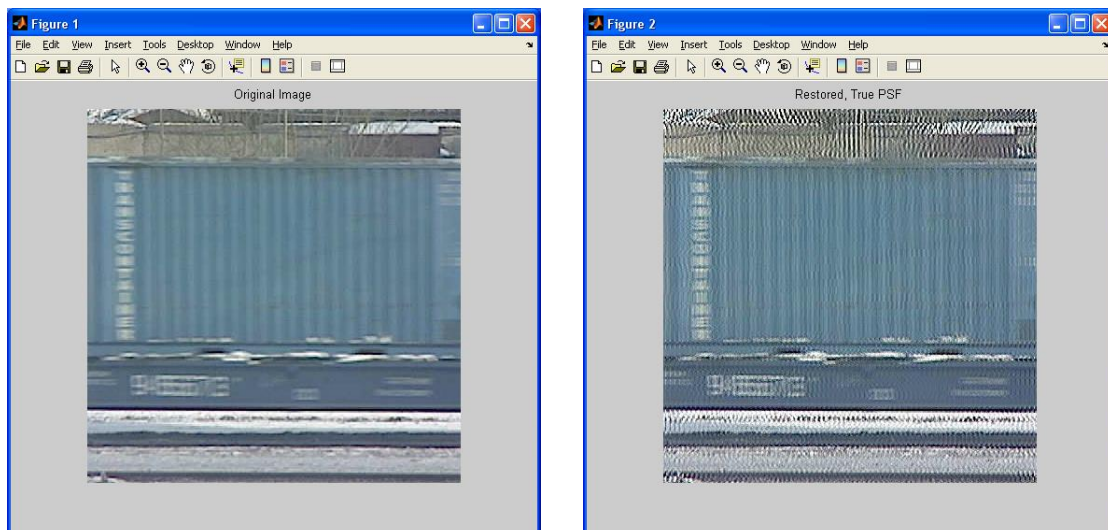
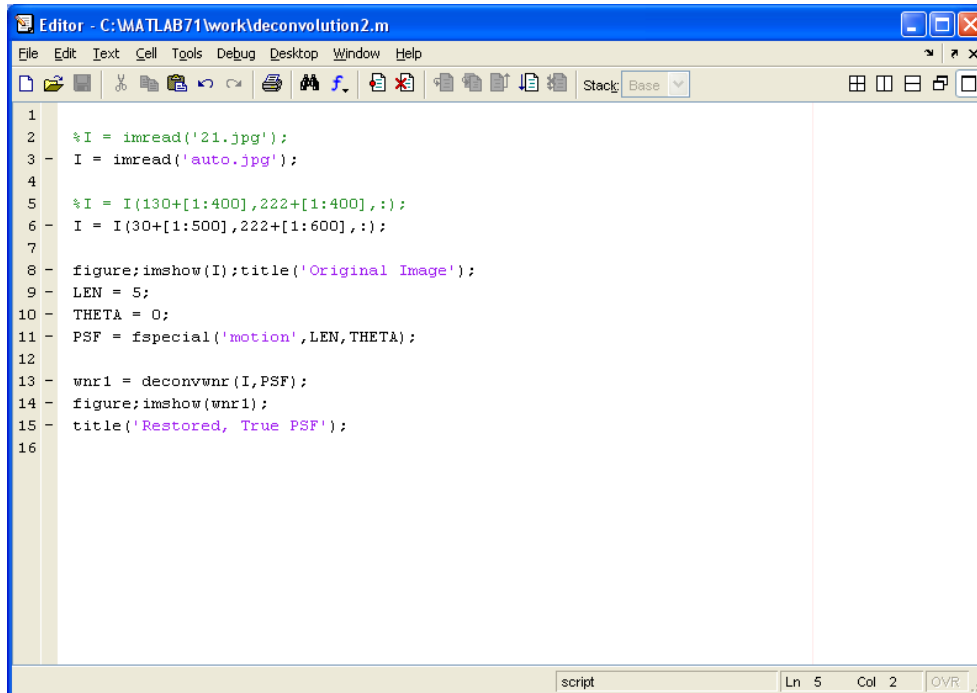


Рисунок 59 – Пример исходного и восстановленного изображения

Изменяя параметр LEN, добейтесь наилучшего восстановления текстовой информации на изображении вагона.

Приведите примеры скриншотов восстановленного изображения.
Откройте файл `deconvolution2.m` (рис. 60). От предыдущего он отличается другой открываемой видекартинкой с другими параметрами.



```
1
2 %I = imread('21.jpg');
3 I = imread('auto.jpg');
4
5 %I = I(130+[1:400],222+[1:400],:);
6 I = I(30+[1:500],222+[1:600],:);
7
8 figure;imshow(I);title('Original Image');
9 LEN = 5;
10 THETA = 0;
11 PSF = fspecial('motion',LEN,THETA);
12
13 wnri = deconvwnr(I,PSF);
14 figure;imshow(wnri);
15 title('Restored, True PSF');
16
```

Рисунок 60 – Окно программы `deconvolution2.m`

Путем подбора параметра `LEN` (рис. 61) добейтесь наилучшего восстановления заднего фона – стены здания.

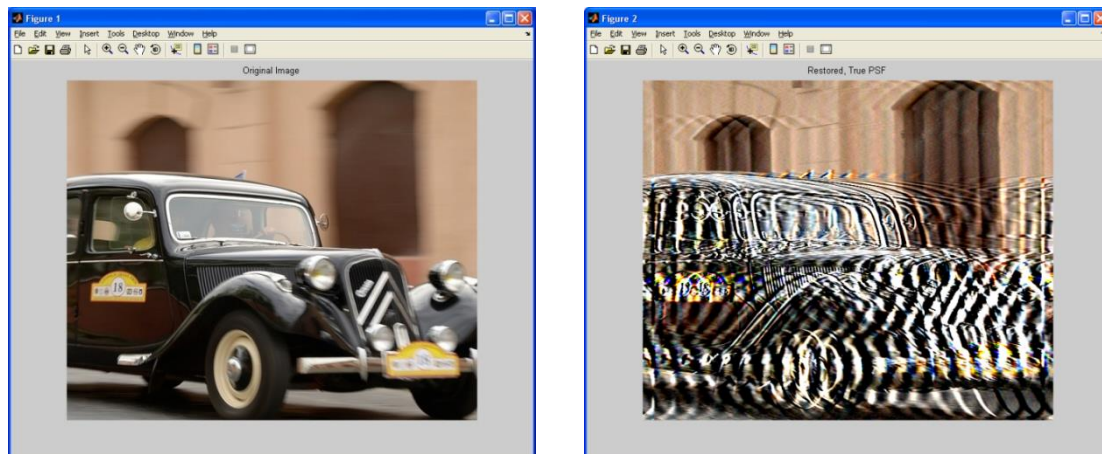


Рисунок 61 – Пример получаемых изображений при выполнении программы `deconvolution2.m`

Оформление отчета

При составлении отчета использовать шаблон с заполненными личными данными студентов.

В отчете привести:

- Цель работы;

- Результаты имитационного моделирования;
- Текст программы deconvolution1.m;
- Основные скриншоты обработки изображения методом деконволюции;
- При каких параметрах движения метод работает плохо;
- Результаты работы с реальным видеосигналом;
- Текст программ deconvolution.m и deconvolution2.m;
- Основные скриншоты;
- Значение параметра LEN, при котором изображение восстанавливается наилучшим образом.

В отчете в свободной форме следует описать характер искажений восстановленного изображения.

Вопросы по работе

1. Каков характер искажения изображения в результате движения объекта в поле зрения?
2. Какие основные методы уменьшения влияния смаза используются на практике?
3. Почему не получается достичь результата восстановления изображения, полученного на практике, по сравнению с математической моделью?
4. Какие искажения изображения, кроме указанного, могут быть уменьшены подобными методами?

5 Лабораторная работа «Изучение телевизионной системы безопасности»

Цель работы

Изучение структуры телевизионной системы безопасности, ее составных частей, приемов работы, настройки оборудования и определения основных характеристик системы [16].

Краткие теоретические сведения

Телевизионная система наблюдения и регистрации (ТСНР) предназначена для работы в системах безопасности и технологического телевидения и представляет из себя аппаратно-программный комплекс, позволяющий осуществлять непосредственное видеонаблюдение, а также запись и воспроизведение видеосигнала с одного или многих источников видеосигналов с одновременной их записью и возможностью воспроизведения записанных видеосигналов на одном или нескольких рабочих местах операторов.

ТСНР состоит из следующих основных частей (рис. 62):

- Источники видеосигналов (телевизионные камеры);
- Усилители-корректоры видеосигналов;
- Видеосерверы;
- Матричный коммутатор;
- Телевизионные мониторы;
- Автоматизированные рабочие места (АРМ) операторов.

Кроме указанного оборудования, в ТСНР обычно присутствуют телевизионные и сетевые кабели для передачи видеосигналов, сетевое оборудование, блоки бесперебойного питания, устройства управления камерами и некоторое другое оборудование.

ТСНР можно разделить на две основные части: подсистему видеонаблюдения и подсистему видеорегистрации, которые объединены функционально на АРМ с помощью программного обеспечения.

Общими устройствами в обеих подсистемах являются телевизионные камеры, усилители, корректоры и АРМ. К подсистеме видеонаблюдения относятся матричные коммутаторы и телевизионные мониторы. К подсистеме видеорегистрации относятся видеосерверы.

В некоторых случаях подсистема видеонаблюдения строится на основе сетевого обмена цифровыми видеосигналами между видеосерверами и АРМ, в этом случае специальное оборудование для подсистемы видеонаблюдения не требуется.

В небольших системах ТСНР может быть построена на базе одного видеосервера, который также выполняет функции АРМ. В этом случае дополнительного оборудования, кроме источников видеосигналов, не требуется.

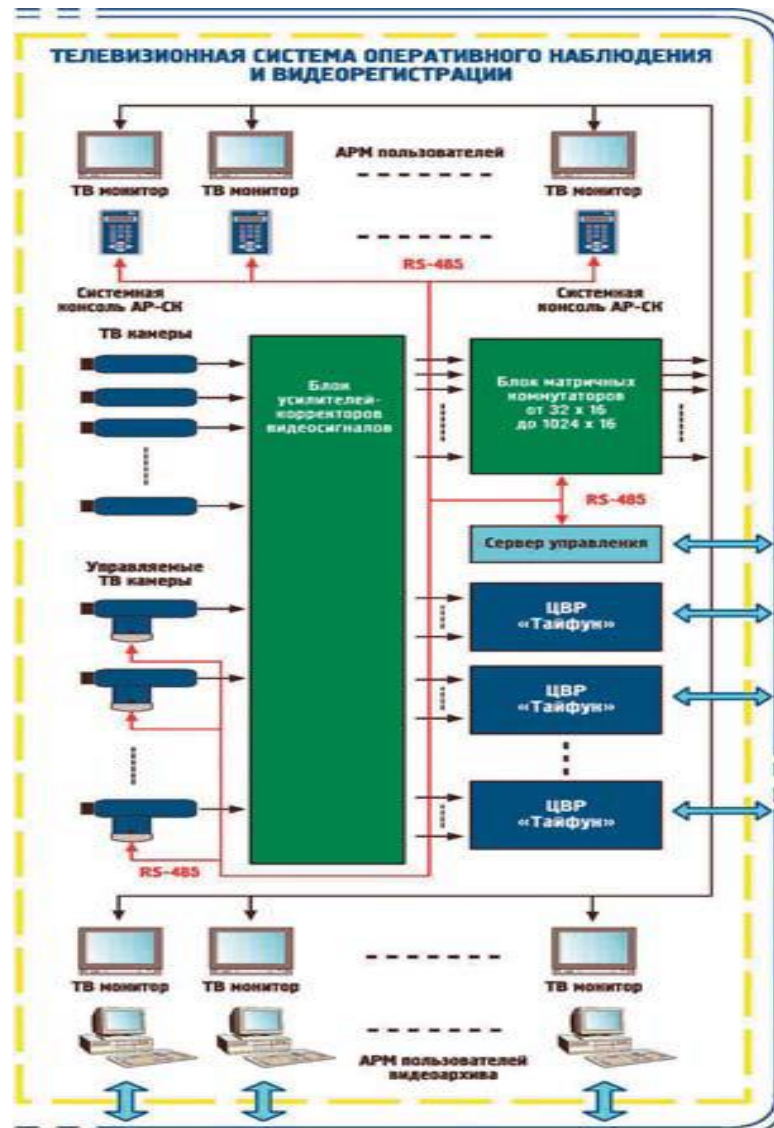


Рисунок 62 – Схема телевизионной системы наблюдения

Программное обеспечение ТСНР обычно позволяет производить масштабирование системы, т.е. на базе одного и того же программного продукта может быть построена как простейшее ТСНР, обслуживающее одну или несколько камер на базе одного системного блока компьютера, так и сложное ТСНР, построенное на базе функционально разделенных системных блоков и АРМ, выполняющее задачи в соответствии с разрешенными действиями того или иного пользователя системы.

Функциональные возможности установки весьма велики, и описание ее достаточно объемно. В данных методических указаниях используется краткое описание, которое позволяет изучить основные приемы работы с системой.

Описание лабораторной установки

Лабораторная установка состоит из персонального компьютера класса Pentium4 с жестким диском 200 Гбайт и установленных в PCI слоты одной или нескольких плат видеозахвата.

Плата видеозахвата представляет собой специализированное устройство на микросхеме Conexant Vt878A.

Это устройство представляет собой аналого-цифровой преобразователь, разработанный специально для обработки цветных видеосигналов и преобразования их в один из стандартов цифрового видеосигнала. Кроме того, устройство имеет встроенный контроллер шины PCI, позволяющей легко подключиться к шине персонального компьютера. Плата имеет входной аналоговый коммутатор на четыре видеовхода, что позволяет подключить к плате до четырех аналоговых источников видеосигнала. Таких плат в системном блоке установлено две. Таким образом, общее число видеовходов составляет восемь.

Система построена на базе макета промышленного или складского здания и включает в себя четыре камеры охраны периметра и одну внутреннюю камеру.

Четыре камеры охраны периметра подключены к четырем входам первой платы видеозахвата, внутренняя камера – к первому входу второй платы видеозахвата.

Ход работы

Работа рассчитана на 4 часа и включает в себя следующие разделы:

- Включение лабораторной установки;
- Изучение основных функциональных элементов программы «Тайфун»;
- Создание собственной конфигурации ТСНР с помощью встроенных программных средств;
- Выключение лабораторной установки.

1. Включение лабораторной установки

Внимание! Лабораторная установка может включаться только преподавателем или лаборантом, или под их непосредственным наблюдением!

Для включения лабораторной установки следует включить блок питания телевизионных камер. Затем включить компьютер с помощью кнопки на его передней стенке.

После включения компьютера автоматически запустится операционная система Windows и автоматически запустится программа «Тайфун».



Рисунок 63 – Программа «Тайфун»

2. Изучение основных функциональных элементов программы

Программа может иметь разнообразный вид на экране компьютера, но состоит она из нескольких основных окон.

Основное окно программы появляется всегда, даже когда программа запущена первый раз и не настроена.

При просмотре пунктов меню не следует выполнять тех действий, смысл которых вы не понимаете.

С помощью преподавателя или лаборанта приведите систему в исходное состояние. При этом на экране монитора будет только изображение окна управления (рис. 64).

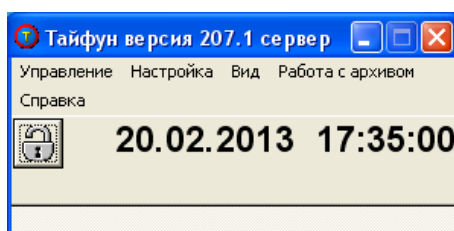


Рисунок 64 – Изображение окна управления

В меню «настройка» выберите пункт «панель инструментов». Появится окно панели инструментов (рис. 65).

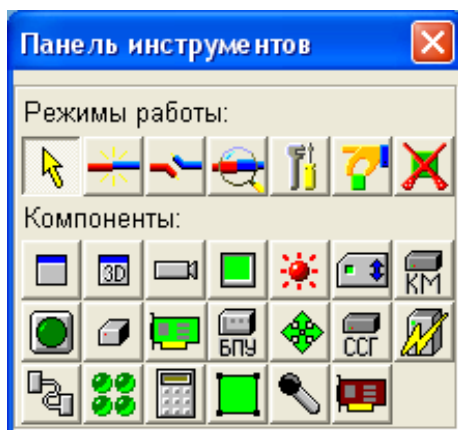


Рисунок 65 – Окно панели инструментов

Необходимо создать окно, на котором будет располагаться план объекта, камеры и другое оборудование. Для этого нужно выбрать кнопку «режим установки и перемещения компонентов системы» и затем нажать кнопку «окно плана помещений»

При этом появится пустое окно.

Необходимо ввести план объекта. В нашем случае им является план стенда складского помещения. Для этого нужно щелкнуть по пустому окну (сделать его активным) и затем в панели инструментов выбрать кнопку «режим настройки компонентов системы». Затем опять щелкнуть по пустому окну. Появится окно «Настройка параметров окна» (рис. 66).

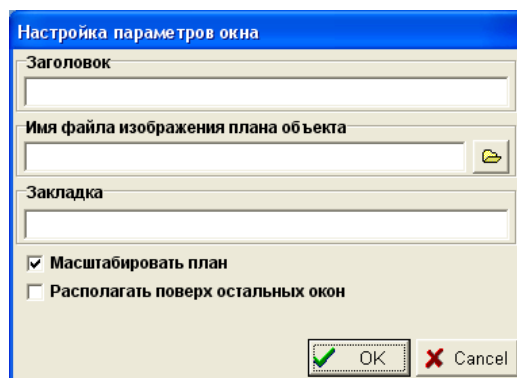


Рисунок 66 – окно «Настройка параметров окна»

Выбрать имя файла изображения плана объекта, для этого можно щелкнуть по изображению папки и в открывшемся списке выбрать файл «стенд.bmp» (рис. 67).

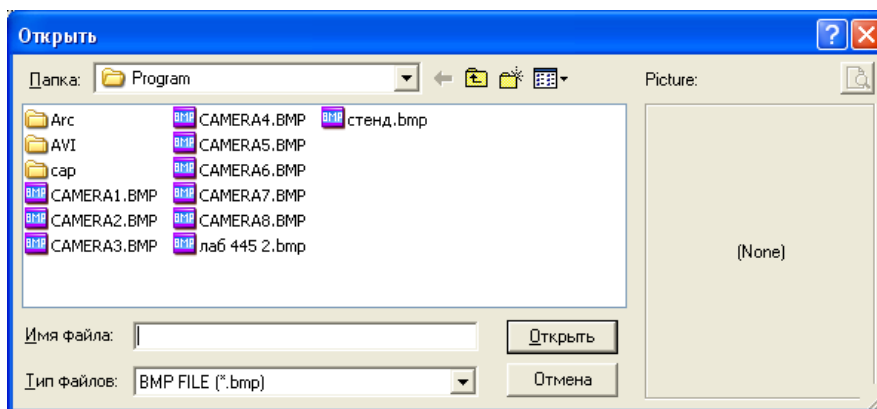


Рисунок 67 – Выбор плана объекта

Изображение плана появится в окне (рис. 68а). Далее это окно следует увеличить примерно до размера 12×8 см (на мониторе 22 дюйма). Для удобства дальнейшей работы окно следует переместить сразу под окном управления.

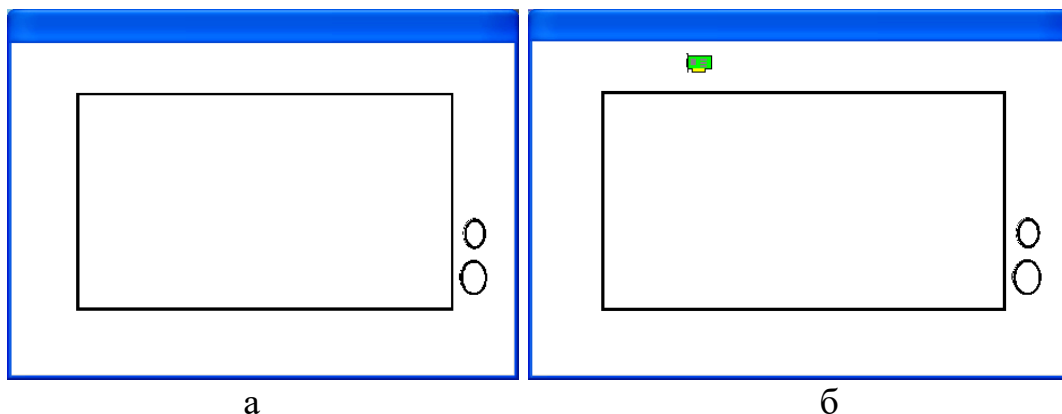


Рисунок 68 – Окно плана объекта

Теперь на плане следует разместить используемое в системе оборудование. Наша система видеонаблюдения состоит из пяти камер и двух четырехканальных плат видеозахвата. При этом четыре первых камеры подключены к четырем входам первой платы, а пятая камера – к первому входу второй платы.

Устанавливаем на плане платы видеозахвата. Для этого в панели инструментов выбираем кнопку «режим установки и перемещения компонентов системы» и затем нажимаем кнопку «Карта ввода изображения». Устанавливаем изображение платы где-нибудь за пределами изображения объекта (рис. 68б).

Теперь нажимаем кнопку «режим настройки компонентов системы» и щелкаем по изображению платы. Открывается окно «Настройка параметров устройства ввода» (рис. 69).

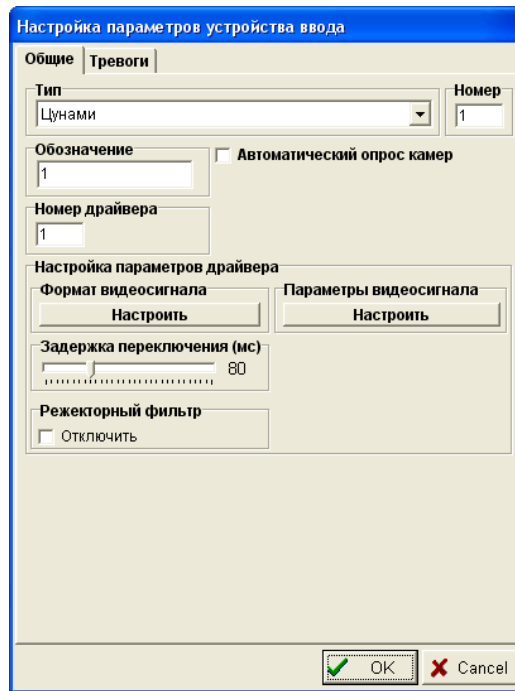


Рисунок 69 – Окно «Настройка параметров устройства ввода»

Прежде всего проверяем номер устройства, драйвера и обозначение – должно быть 1. Теперь в списке устройств выбираем тип платы – Цунами. Именно такие платы установлены в компьютере. Далее нажимаем кнопку Формат видеосигнала – настроить. Появляется окно настройки (рис. 70).

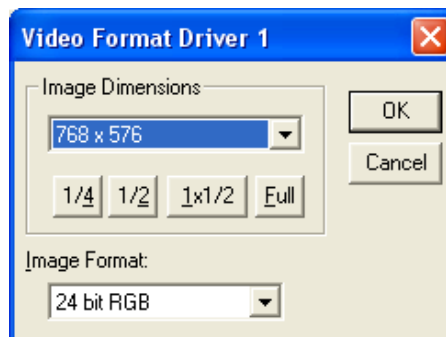


Рисунок 70 – Окно настройки

Выбираем формат кадра 768×576 – максимально возможный для камер стандартного разрешения.

После этого устанавливаем на плане вторую плату видеозахвата (рис. 71) и настраиваем ее (см. выше). Плата должна быть 2, но номер драйвера – 1.

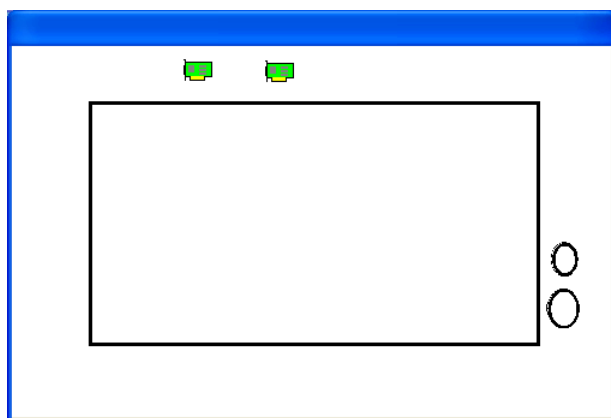


Рисунок 71 – Добавление плат видеозахвата

Устанавливаем камеры (рис. 72). Для этого следует нажать кнопку «режим установки и перемещения компонентов системы» и затем кнопку «ТВ камера». Устанавливаем изображение камеры в месте, примерно соответствующему реальному расположению камер на объекте. Устанавливаем таким образом все 5 камер.

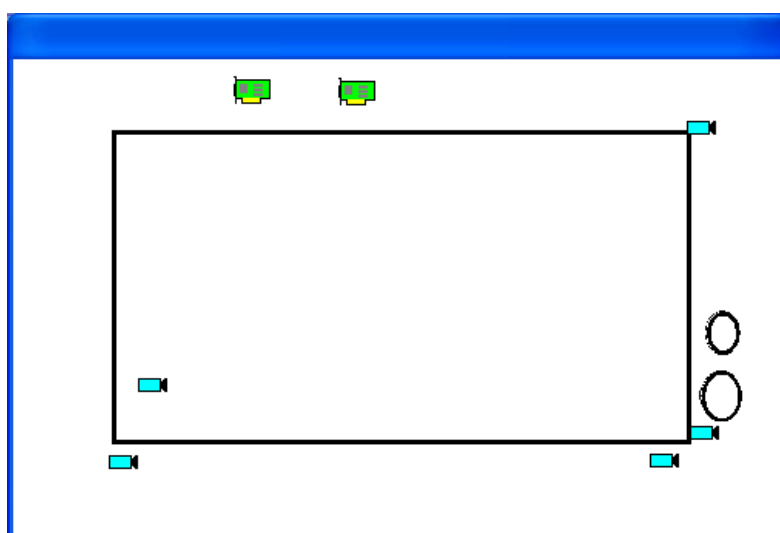


Рисунок 72 – Установка камер

Теперь необходимо выбрать правильное изображение камер, для чего выбираем пункт «Режим настройки компонентов системы» и затем щелкаем по изображению первой камеры. Открывается окно «Настройка параметров ТВ-камеры» (рис. 73).

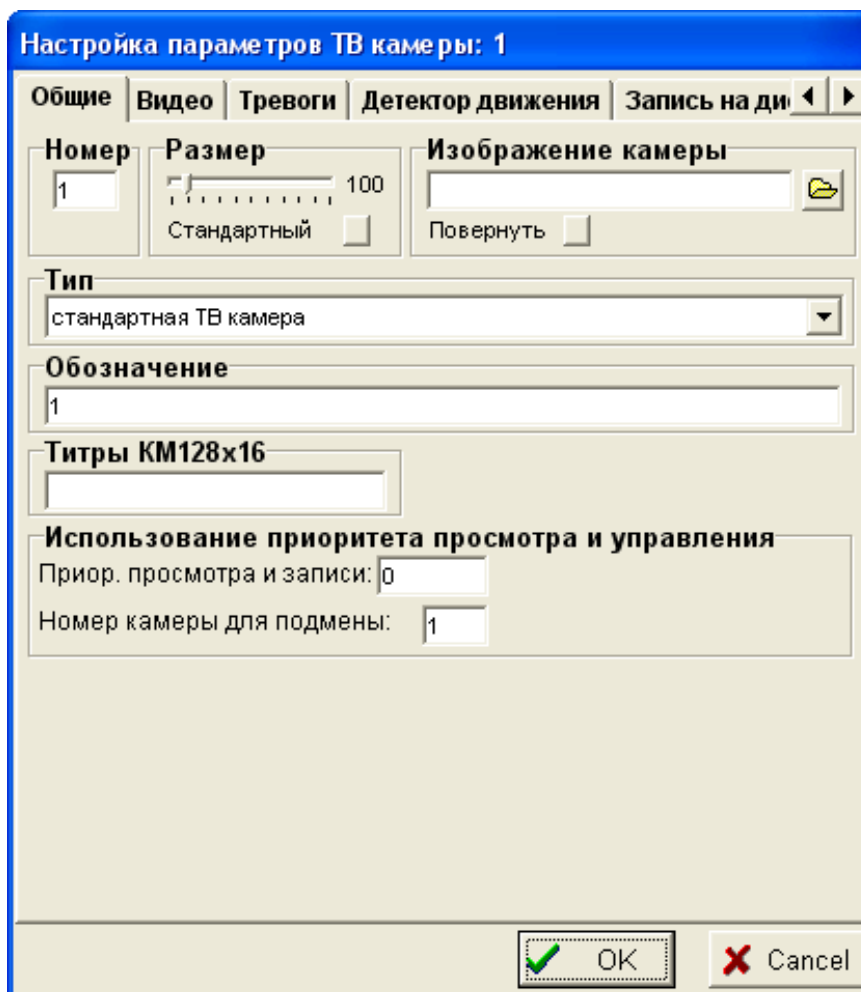


Рисунок 73 – Окно «Настройка параметров ТВ-камеры»

Выбрать нужное изображение камеры можно, щелкнув по изображению папки, и в открывшемся окне выбрать реальное направление визирования камеры (рис. 74).

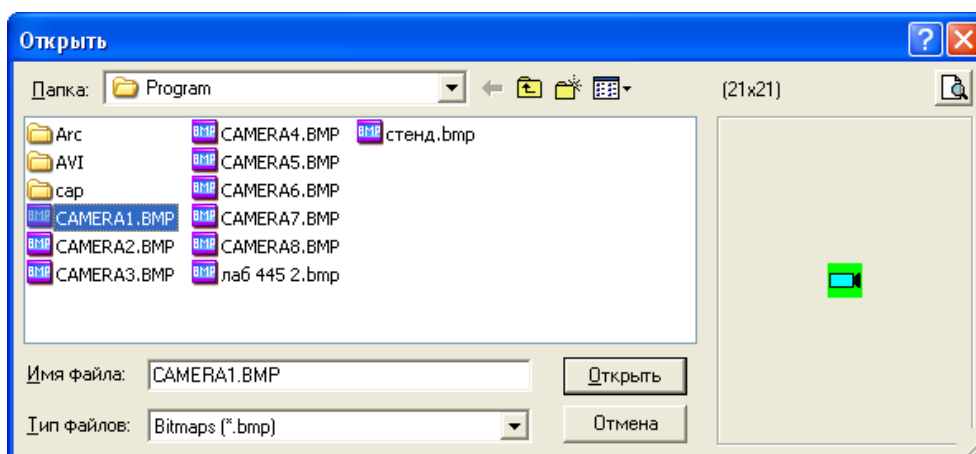


Рисунок 74 – Выбор направления визирования камеры

Аналогичные действия следует выполнить со всеми оставшимися камерами (рис. 75).

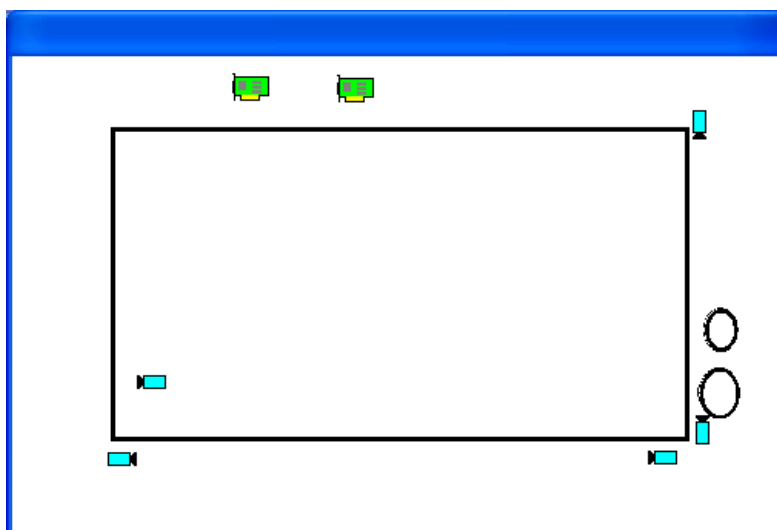


Рисунок 75 – Окно программы после выбора направления визирования каждой камеры

Далее необходимо выполнить логическое соединение оборудования в системе так, чтобы оно соответствовало физическому подключению. Для этого следует выбрать на панели инструментов кнопку «Режим установки соединений», затем щелкнуть по изображению камеры 1 и в открывшемся окне выбрать единственный пункт «Выход видео», подтвердить кнопкой ОК (рис. 76).

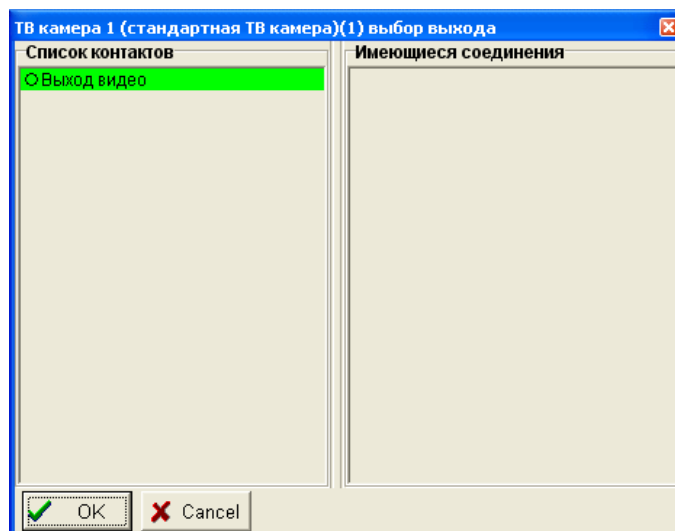


Рисунок 76 – Установка логического соединения оборудования

После этого щелкнуть по изображению первой платы видеозахвата. Откроется окно соединений платы. Необходимо выбрать «вход видео1» и нажать ОК. Появится короткое сообщение «Соединение добавлено».

Необходимо выполнить эти действия для камер 2,3,4 и первой платы, а также для камеры 5 и второй платы.

После выполнения этих действий соединения можно проверить, нажав кнопку на панели инструментов «Режим просмотра соединений» и,

например, щелкнув по изображению платы 1. Появится ее таблица соединений (рис. 77).

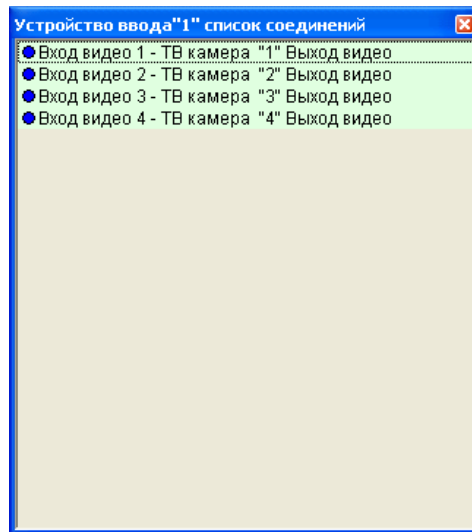


Рисунок 77 – Таблица соединений

Аналогично можно выполнить такую проверку для всех элементов системы.

При правильных соединениях можно открывать окно просмотра камер. Для этого следует закрыть панель инструментов или нажать на ней кнопку со стрелкой – перевод в рабочий режим. Теперь, щелкнув по изображению первой камеры правой кнопкой мыши и выбрав пункт «открыть окно просмотра», можно получить окно с изображением от первой камеры (рис. 78).



Рисунок 78 – Окно с изображением первой камеры

Выполнив те же действия со всеми остальными камерами, получим пять окон просмотра. Необходимо их выставить так, чтобы на экране их удобно было бы наблюдать (рис. 79).

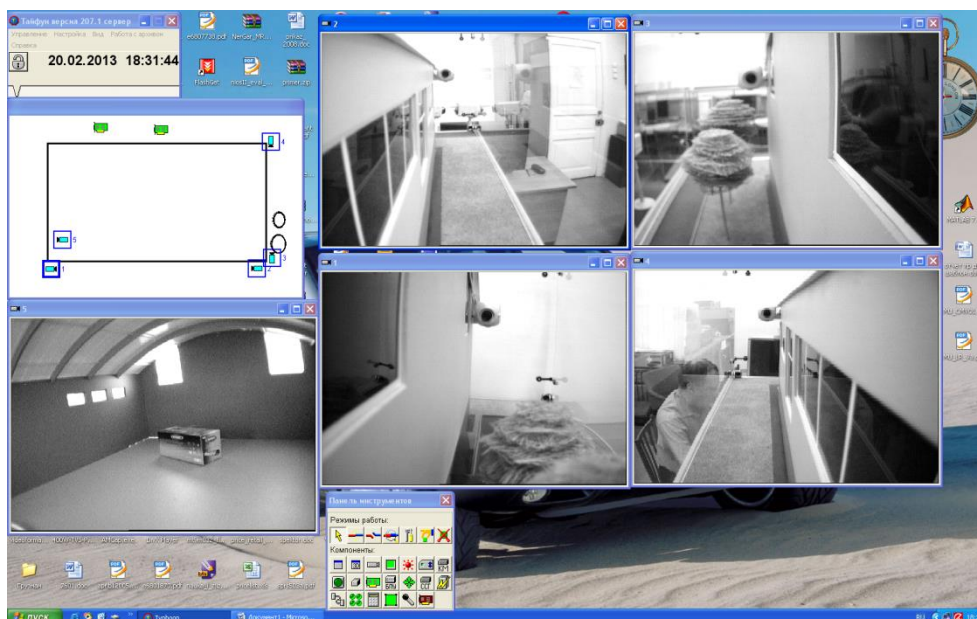


Рисунок 79 – Окно с изображениями со всех камер

Теперь необходимо включить режим записи для каждой камеры. Например, для камеры 1 можно щелкнуть по ее окну правой кнопкой и выбрать пункт «включить непрерывную запись». В правом верхнем углу должен появиться желтый кружок (рис. 80).



Рисунок 80 – Включение записи камеры

Проверить режим просмотра оперативного архива можно, щелкнув по окну изображения от камеры (например, 1) правой кнопкой мыши и выбрав пункт «просмотр записей». Появится окно проигрывателя (рис. 81).

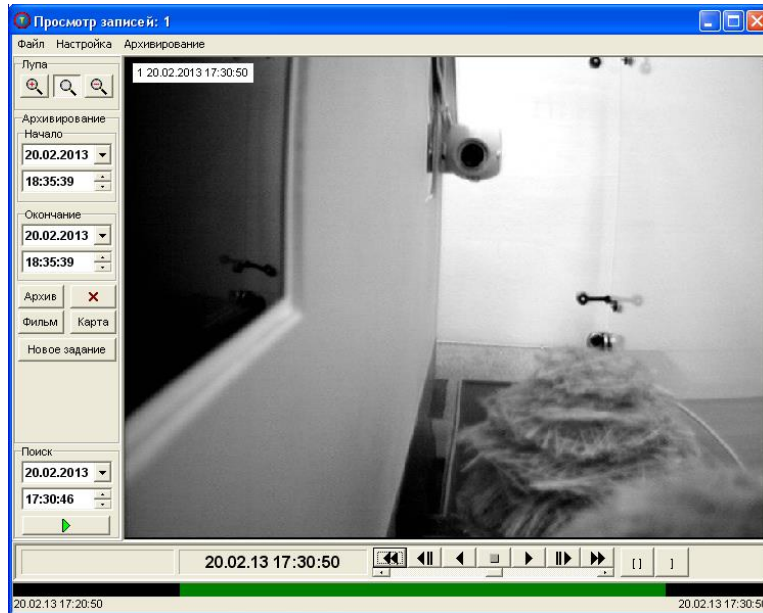


Рисунок 81 – Окно проигрывателя

В проигрывателе имеются следующие средства управления:

- традиционные кнопки управления медиаустройством (старт, стоп, ускоренный и обратный просмотр),
- выбор камеры для воспроизведения,
- выбор времени или временного интервала,
- поиск событий по линейке активности.

Используя кнопки управления воспроизведением, просмотрите фрагменты записей. Обратите внимание, что при наведении на окно воспроизведения курсора и нажатии правую кнопку мыши, активизируется режим экранной лупы.



Рисунок 82 – Окно программы с воспроизведением нескольких видеосигналов

В исследуемой системе присутствует также дополнительная важная функциональная возможность, связанная с воспроизведением одновременно нескольких видеосигналов параллельно.

Откройте окно проигрывателя. Для этого наведите курсор мыши на изображение от выбранной камеры. Щелкните правой кнопкой мыши и выберите пункт «Просмотр записей» (рис. 82).

Изучите назначение органов управления и возможности просмотра архивных видеозаписей.

4. Выключение лабораторной установки

Внимание! Лабораторная установка может выключаться только преподавателем или лаборантом, или под их непосредственным наблюдением!

Последовательность выключения установки обратна ее включению.

Оформление отчета

Отчет должен быть оформлен в электронном виде и распечатан. За основу отчета должен быть взят прилагаемый шаблон lr1.dot. В имеющемся шаблоне следует заполнить обязательные поля, включающие ФИО и группу студента, дату выполнения работы, измеренные параметры видеосигнала по п.п. 2-4, а также скриншоты основных окон программы.

Размер отчета должен быть не более трех страниц. При необходимости вставленные в шаблон рисунки следует отмасштабировать.

Вопросы по работе

1. Какие основные элементы входят в состав телевизионной системы безопасности?
2. Какие основные элементы программного интерфейса телевизионной системы безопасности реализованы в программе «Тайфун».
3. Для чего нужно и как реализована компрессия цифрового видеосигнала в телевизионной системе безопасности?
4. Каковы основные этапы настройки программы телевизионной системы безопасности?

Список используемых источников

1 Ярышев С.Н., Сычева Е.А. Методические указания по выполнению лабораторных работ по курсу Видеоаналитика и цифровая обработка видеосигнала: Учебно-методическое пособие - Санкт-Петербург: СПб: Университет ИТМО, 2017, 2017. - 92 с. - экз.

2 Скамья оптическая MXT120-030-12. Электронный ресурс. URL: <http://www.jxlszb.com>

3 Плата MXT030-025-12 Электронный ресурс. URL: <http://www.jxlszb.com>

4 Микрометрическая подвижка LSSP-50JV Manual Translation Stage Электронный ресурс. URL: <http://www.jxlszb.com>

5 Держатель трафарета Adjustable plate holder LSGB2-7 Электронный ресурс. URL: <http://www.jxlszb.com>

6 Стойка монтажная LSZG1-150 Электронный ресурс. URL: <http://www.jxlszb.com>

7 Осветитель лабораторный Электронный ресурс. URL: <http://www.lomo-eltem.ru>

8 Сетевой коммутатор. Электронный ресурс. URL: <http://osnovo.ru>

9 Трехкоординатная микрометрическая подвижка LSSZ-1003 Электронный ресурс. URL: <http://www.jxlszb.com>

10 Right angle brackets LSZJ2 Электронный ресурс. URL: <http://www.jxlszb.com>

11 В.П. Дьяконов. Matlab 6.5 SP1/7 + Simulink 5/6. Работа с изображениями и видеопотоками. М.: СОЛОН-Пресс, 2005. - 400с.

12 Цветная мегапиксельная телевизионная камера высокого разрешения в корпусе внутреннего исполнения Модель VEC-545-USB. Электронный ресурс. URL: http://evs.ru/d_sheet/VEC-545-USB.pdf.

13 Метод создания сферических панорам из изображений, полученных всенаправленными оптико-электронными системами В.П. Лазаренко, Т.С. Джамийков, В.В. Коротаев, С.Н. Ярышев.

14 Использование метода surf для обнаружения устойчивых признаков изображения при создании сферических панорамных снимков Г. М. Джгаркава, Д. Н. Лавров КиберЛенинка: <https://cyberleninka.ru/article/n/ispolzovanie-metoda-surf-dlya-obnaruzheniya-ustoychivyh-priznakov-izobrazheniya-pri-sozdanii-sfericheskikh-panoramnyh-snimkov>.

15 Пример высококачественной панорамы Москвы, полученной с Останкинской телебашни. Электронный ресурс. URL: <http://www.airpano.com/files/Moscow-Giga-Chistoprudov-cl/1-2>

16 Цифровая телевизионная система наблюдения и регистрации «тайфун» руководство системного администратора ЭВС1.132.001-РА 2003 г. ООО ЭВС www.evs.ru.

17. Шлихт Г.Ю. Цифровая обработка цветных изображений.-М., Издательство ЭКОМ, 1997. – 336 с.

**Сергей Николаевич Ярышев
Елена Александровна Сычева
Алексей Александрович Горбачев**

**Видеосистемы безопасности.
Методические указания по выполнению
лабораторных работ**

Учебно-методическое пособие

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверкский пр., 49