

МИНОБРНАУКИ РОССИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
“Санкт-Петербургский государственный электротехнический
университет “ЛЭТИ” им. В.И. Ульянова (Ленина)”

Факультет электроники

Направление № 210 100.68 «Электроника и нанoeлектроника»

Магистерская программа

Кафедра микро- и нанoeлектроники

К защите допустить:

Заведующий кафедрой

(д.т.н., доц.)

/ В.В.Лучинин /

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

***Тема: Разработка миниатюрного блока распознавания
микроколоний бактерий для лаборатории на чипе***

Студент

/Каблуков Д.Е./

Руководитель

(Учёная степень, учёное. звание)

/В.В. Лучинин/

Руководитель магистерской

программы

(Учёная степень, учёное. звание)

/В.В. Лучинин/

Консультанты

(Учёная степень, учёное. звание)

/А.В. Соловьёв/

Санкт-Петербург
2014 г.

Факультет Электроники
Направление 210100. 68 «Электроника и нанoeлектроника»
Магистерская 210161.68 «Нанотехнология и диагностика» ИЛИ
программа 210162.68 «Нанoeлектроника и фотоника» ИЛИ
 210163.68 «Нано- и микросистемная техника»
Кафедра Микро- и
 нанoeлектроники

УТВЕРЖДАЮ
Заведующий
кафедрой МНЭ

д.т.н. В.В.Лучинин

«_____» _____ 201__ г.

ЗАДАНИЕ на магистерскую диссертацию

Студент _____ Группа № _____

Место выполнения диссертации _____

1. Тема диссертации _____

2. Исходные данные /технические требования _____

3. Содержание диссертации _____

4. Перечень отчётных материалов _____

5. Дополнительные разделы _____

6. Дата представления работы к защите _____

Руководитель
(Учёная степень, учёное звание) _____ / _____ /

Студент _____ / _____ /

«_____» _____ 201__ г.

КАЛЕНДАРНЫЙ ПЛАН

выполнения магистерской диссертации

№ п/п	Наименование работ	Срок выполнения	Отметка о выполнении
	Предварительное рассмотрение работы (предварительная защита)		
	Представление работы к защите	(см. Задание)	

Руководитель

(Учёная степень, учёное звание)

/ /

Студент

/ /

СОДЕРЖАНИЕ

Введение.....	8
Глава 1. Обзор информационных источников	
1.1 Лаборатории на чипе для повышения эффективности биомедицинской диагностики.....	11
1.2 Методы культивации или посева.....	16
1.3 Ювенильные или микро- колонии. Особенности распознавания.	24
1.4 Различные методы распознавания образов. Метод шаблона. Пространство признаков.	27
1.5 Обзор программных реализаций методов анализа изображений для биомедицинских приложений.	31
1.6 Open CV для Android. Обзор.....	32
1.7 Постановка задачи дипломного проекта.....	34
Глава 2. Разработка программного обеспечения для системы распознавания образов.	
2.1 Получение базы данных контрастных изображений..... колоний.	36
2.2 Цифровая фильтрация изображений: выбор фильтра для подавления шума.....	37
2.3 Выделение контуров колоний.....	41
2.4 Выделение признаков для распознавания микробиологических объектов.....	47
Глава 3. Экспериментальная часть (Миниатюризация блока видеорегистрации)	
3.1 Выбор оптической системы.	53

3.2 Реализация оптической системы (расчет).....	61
3.3 Выбор программной среды (оболочки) для программирования на мобильных устройствах.....	62
3.4 Адаптация мобильного устройства для его использования в качестве микроскопической камеры.....	65
Глава 4. Результаты и обсуждение.....	77
Заключение.....	88
Список используемой литературы.....	89
Приложения.	
Приложение 1.....	91

Реферат

Пояснительная записка содержит: 111 с., 46 рис., 7 табл., 21 источник,
1 прил.

Ключевые слова: *лаборатория на чипе, распознавание образов, ювенильные микроколонии, микроскоп, геометрические и цветовые признаки объектов.*

Объектом исследования являются ювенильные микроколонии и методы их распознавания при помощи компьютерных методов и методов на основе мобильных платформ.

Цель работы – разработка алгоритма распознавания образов на персональном компьютере и смартфоне на ОС Android, а также миниатюризация системы компьютерного зрения.

В результате работы были определены признаки бактерии для идентификации, реализованы алгоритмы распознавания, а также рассчитана и изготовлена система для компьютерного зрения.

Эффективность полученного алгоритма подтверждается проведенными экспериментальными исследованиями.

Annotation

Solutions of problems in the field of digital image processing generally require extensive experimental work involving software simulation and testing with large sets of sample images. Although algorithm development typically is based on theoretical underpinnings, the actual implementation of these algorithms almost always requires parameter estimation and, frequently, algorithm revision and comparison of candidate solutions. Thus, selection of a flexible, comprehensive, and well-documented software development environment is a key factor that has important implications in the cost, development time, and portability of image processing solutions.

ВВЕДЕНИЕ

В конце XX века общество стало серьезнее относиться к мониторингу параметров организма человека. Каждый из нас хотя бы раз в жизни сдавал, анализ крови. При этом ждать результатов приходилось ждать очень долго – ведь сначала он отправлялся в лабораторию, там анализировался, затем отправлялся в виде распечатки врачу. Помимо временных затрат также были и финансовые, так как врачам, лаборантам надо платить деньги, также аренда помещения и использование дорогих приборов также добавляют свою цену. А ведь порой скорость, точность и доступность анализа – вопрос жизни и смерти в прямом смысле слова.

И как было бы здорово, если бы врачи или даже сами пациенты могли мгновенно проводить сложнейшие анализы и получать результаты в течение нескольких минут! Представьте какой бы удобной была лаборатория если уменьшить её до размеров микрочипа, а все производимые в ней человеческие действия по доставке, перемещению и анализу образца полностью автоматизировать!

А теперь представьте, что такие лаборатории уже существуют! Называются они лабораториями на чипе (от англ. lab*on*chip). Один чип размером порядка 4х4 см может заменить целый комплекс оборудования, необходимого для анализа ДНК/РНК, установления родства, определения генетически модифицированных организмов, ранней диагностики онкологических заболеваний, изучения эффективности трансфекции клеток, количественного определения белков, определения уровня экспрессии генов и многого другого!

Главными достоинствами микроаналитических систем, определяющими их высокий аналитический потенциал по сравнению с уже известными системами анализа, являются:

1. Возможность интеграции многих аналитических компонентов и их дублирование (многоканальные системы) в одном чипе.
2. Сверхэффективность и скоростной производительный анализ.

3. Секвенирование ДНК (десятки тысяч проб в сутки). Суперфракционирование без сорбентов.

4. Гидравлическая логика для безнасосной подачи реагентов, ввода пробы.

5. Универсальность масштабов сложности: от одноканального до 100-канального анализатора для сверхпроизводительного секвенирования ДНК, лабораторий на чипе.

6. Возможность альтернативных путей решения аналитических задач. Перспектива создания сэндвич-сенсоров одноразового пользования.

7. Сверхвысокая чувствительность детектирования за счет использования лазер-индуцированной флюоресценции.

8. Заимствованные из микроэлектроники высокопроизводительные автоматизированные технологии разработки и изготовления капиллярно-матричных микрочипов (компьютерное проектирование, фотолитография, технологии LIGA — глубокая рентгеновская литография, микролитье пластиковых чипов, неограниченное масштабирование производства на единой технологической основе), что обеспечивает быстрый переход от разработки к массовому выпуску — в течение месяца — специализированных приборов для конкретных целей.

9. Беспрецедентные возможности усовершенствования иммуноанализа позволяют детектировать одномолекулярную иммунореакцию, повысить производительность иммуноанализа до рекордных величин - сотен тысяч определений в сутки.

К настоящему времени рост числа разработок в области микроаналитических систем, появление промышленных стандартов, например, ISO показывает, что это направление отвечает современным потребностям общества.

Тем не менее, инженерные и промышленные реализации микроаналитических систем все еще достаточно ограничены. Это объясняется новизной тематики, в которой еще не сформировались типовые решения и технологии, стандарты,

комплектующие и другие аспекты среды разработки инновационных изделий, имеющиеся в развитых индустриальных обществах. Каждая новая разработка микроаналитической системы является научным изобретением. Например, наиболее широко используемый в клинической аналитике формат – это стандартная 96 ячеечная биохимическая платформа – планшет. Анализ в таком планшете осуществляется, как правило, ручным методом. Имеются и приборы для считывания результатов анализа в платформе. Но все такие установки громоздки и имеют низкую точность и скорость анализа.

Простые чипы можно производить промышленными масштабами с небольшими затратами, что позволяет распространить их в бедные районы земного шара по заметно сниженным ценам по сравнению с ценами, которые предлагают фирмы, производящие данные чипы дорогостоящими методами. Об упрощении и удешевлении производства, а также об основах анализа данных и будет эта дипломная работа

Глава1 Обзор информационных источников

1.1 Лаборатории на чипе для биомедицинской диагностики

В нашей стране постепенно и повсеместно утверждаются приборы для анализа ДНК. В своем большинстве это достаточно крупные приборы весом в несколько килограмм, для которых требуются отдельные боксы и помещения в диагностических лабораториях. В то же время технология на Западе не стоит на месте. Последние 15 лет характеризовались настоящей биотехнологической революцией в плане миниатюризации устройств для анализа ДНК. Лаборатория на чипе - термин на английском lab-on-a-chip. Синонимы - микросистемы полного анализа, micro total analysis systems. Аббревиатуры - LOC, μ TAS. Лаборатория на чипе - миниатюрный прибор, позволяющий осуществлять один или несколько многостадийных (био)химических процессов на одном чипе площадью от нескольких мм^2 до нескольких см^2 и использующий микро- или наноскопические количества образцов для пробоподготовки и проведения реакций.

Недавно вышла статья в журнале International J. Neurology. В ней группа исследователей под руководством доктора Нэ Юн Ли из Института Нанобиотехнологий Южной Кореи представила свой вариант микроприбора из класса Lab-on-a chip (LOC) и продемонстрировала потенциальное использование микрофлюидного устройства этого типа для задач клинической диагностики.

Это современная "ноу-хау"-технология с применением полимерных материалов (например, полидиметилсилоксана) для миниатюризации обычных лабораторных приборов. В конкретике, было показано микромасштабирование полимеразной цепной реакции (ПЦР) с использованием одного нагревателя, микроустройства для инъекций образца с одноразовых пластиковых шприцев и стратегия сборки устройства с использованием простых методов модификации.

"Мы стремимся построить полностью интегрированную одноразовую микрожидкостную систему с простым управлением, что удобно для использования в клинике для молекулярной диагностики", - заявляет Нэ Юн Ли.

С ростом интереса к методам миниатюризации устройств подобные платформы LOC разрабатываются для применения в различных областях исследований, таких как аналитическая химия, биохимические анализы, экологические тесты. Для использования в клинике критичными являются следующие критерии: быстрота реакции, малые объемы образца и желательно одноразовые материалы для предотвращения загрязнения. Низкая стоимость также важна для широкого применения микрожидкостных устройств. В частности, огромное количество научной продукции было произведено в области микромасштабирования ПЦР.

В первые дни основным выбором для создания устройств были такие материалы как кремний, стекло и кварц. Однако, на основе кремния микроустройства стоят дорого и требуют сложных процессов изготовления. Они также являются трудновоспроизводимыми и хрупкими. Эти недостатки привели многих исследователей к поиску лучших материалов с точки зрения простоты изготовления, производства стоимости и надежности устройства. В результате нашли свое применение различные пластиковые материалы, например, полидиметилсилоксан (PDMS), полиметилметакрилат (ПММА), полиимид (PI) и полиэтилен (ПЭТ).

"В дополнение к выбору оптимального материала для изготовления микроустройства, каждая единица из компонентов LOC- системы должна работать в едином режиме. LOC обычно состоит из нескольких компонентов блока, например, пробоподготовки, разделения и системы детекции, каждый из которых имеет свой оптимальный режим работы. К примеру, механизм пробоподготовки, в основном состоящий из ДНК экстракции и очистки клеток, может быть реализован с помощью пневматических, центробежных и магнитных сил. В результате интеграция всей системы в единый монолитный механизм никогда не бывает простой задачей", - говорит Нэ Юн Ли. Реальным примером

подобной технологии могут служить продукты ведущих в этой области компаний Affymetrix (“GeneChip”) или Agilent (“LabChip”), производящих лаборатории на чипе для генетических анализов. В таких чипах ДНК анализируется методом полимеразной цепной реакции (ПЦР). Его суть заключается в последовательном нагревании и охлаждении раствора, содержащего образец анализируемой ДНК, два праймера (участки ДНК в 20 нуклеотидов, комплементарные участкам анализируемого ДНК), смесь четырех нуклеотидов и фермент ДНК-полимераза. При нагревании двунитевые отрезки ДНК расщепляются; при охлаждении под действием фермента каждая из однонитевых цепочек достраивается до двунитевой копии исходного отрезка.

Число отрезков молекулы ДНК удваивается при каждом цикле, и из каждой молекулы в исходной пробе через 30 циклов образуется 230 – более миллиарда копий. Метод был изобретен в 1987 г, и в то время ученым приходилось каждые несколько минут вручную переставлять пробирки из одной водяной бани в другую и после каждого цикла добавлять в них новую порцию фермента. Сегодня ту же работу выполняют миниатюрные автоматизированные лаборатории.

Как видим, менее чем за 20 лет метод ПЦР усовершенствовался примерно так же, как компьютеры – за полвека.

Компания CombiMatrix предложила чип размерами с почтовую марку для определения биологической опасности. Устройство, содержащее такой чип, может определить присутствие нескольких видов микроорганизмов, применяющихся в составе бактериологического оружия. На его базе CombiMatrix выпустила детектор HANAA, который можно использовать в полевых условиях. Прибор помещается в ладони, питается от батареек и весит около одного килограмма. Каждая микроскопическая ячейка чипа заменяет установку для проведения ПЦР. Прибор анализирует ДНК и соотносит с одним из запрограммированных типов патогенной ДНК. Ячейки, в которых присутствует ДНК одного из определяемых чипом патогенных микроорганизмов, флюоресцируют, а их свечение улавливается датчиком. Процесс обработки

четырёх различных образцов занимает 30 минут. Как говорят разработчики прибора, он может опознать патоген при концентрации 10 бактерий в 1 пробе (1 проба представляет собой капсулу диаметром 5 мм и 2 см длиной). CombiMatrix также выпустила устройство на основе чипа, в котором проводится иммуноферментный анализ. В его ячейках светятся антитела к ядам, не содержащим ДНК. Такое устройство может опознавать 5 токсинов типа рицина.

Для производства лабораторий на чипе используются почти те же технологии, что и для производства микросхем, включая литографию и травление. Однако лаборатория на чипе, в отличие от планарной микросхемы, должна быть трехмерной. Причина заключается в том, что если электричество может протекать по плоскому проводу, то жидкость не течет по сплюснутому шлангу. Таким образом, при производстве лабораторий на чипе используются совмещенные методы планарной и MEMS/NEMS-технологий.

Итак, лаборатория на чипе представляет собой МЭМС-устройство для биохимических анализов. Принимая во внимание стремительный прогресс в области МЭМС-систем, можно прогнозировать, что в дальнейшем размеры и стоимость устройств будут уменьшаться (а функциональность, соответственно, расти), и через несколько лет такая лаборатория станет для нас таким же привычным средством диагностики самочувствия, как сегодня термометр.

В таблице 1.1 приведены основные разновидности микросистем для анализа данных.

Таблица 1.1-Основные классы микроаналитических систем

Тип	Класс	Основные принципы
1	Матричные микрочипы (генные, гибридизационные чипы)	1. Топологически кодированный синтез цепных молекул-зондов с помощью многостадийной фотолитографии. 2. Фотоинициирование реакций. 3. Молеклярное распознавание.
2	Капиллярные микрочипы (в том числе с функциональными элемен-тами, например, ПЦР реакторами, химическими реакторами)	1. Капиллярный электроосмоти-ческий транспорт. 2. Микрогидравлика. 3. Микромеханика. 4. Оптоэлектроника.
3	Сверхвысокопроизводительные компьютеризованные скоростные системы адресной доставки проб – платформы.	1. Электроосмотический транспорт. 2. Микрогидравлика.

ПЦР в настоящее время является незаменимым инструментом для анализа и диагноза. Микромасштабные ПЦР значительно сократили общее время анализов вследствие внедрения элементов с повышенной теплоотдачей с быстрым переходом из одного температурного режима в другой. В результате увеличилось отношение поверхности к объему внутри микроканального реактора прибора. Таким образом, общее время реакции амплификации нуклеиновых кислот было уменьшено до 6 минут (в противовес стандартному периоду ПЦР в 90 минут), а объем образца - до 100 нанолитров (по сравнению со стандартными 25 микролитрами).

В проточном ПЦР жидкий образец постоянно движется внутри микроканала, вдоль которого размещены несколько дискретных источников нагрева. Для амплификации нуклеиновых кислот обычно требуются три различные температурные зоны для денатурации, отжига и элонгации. Таким образом, многоступенчатый контроль температуры в ПЦР всегда был ограничивающим фактором в миниатюризации и портативности устройств. До сих пор исследователи придумывали различные источники тепла, например, блоки нагрева металла, тонкие пленочные обогреватели, инфракрасные устройства, галогенные лампы, микроволновые печи, платиновые резисторы при выполнении

ПЦР на чипе. Тем не менее, использование нескольких источников нагрева увеличивает общие габаритные размеры устройства.

"Мы значительно упростили вопрос регулирования температуры путем изготовления микроустройства с помощью укладки его в несколько слоев. Было изготовлено трехмерное (3D) прямоточное ПЦР-микроустройство путем сборки двух PDMS-субстратов в вертикальном расположении", - рассказывает Нэ Юн Ли.

В то же время следует отметить, что миниатюрным может являться только собственно микрочип, т. е. рабочий модуль, куда подается анализируемая проба, а устройства, считывающие с чипа результат анализа, могут существенно превышать по своим габаритам сепарационную систему.

1.2 Методика культивации бактерий и грибов.

1.2.1. Приготовление питательной среды.

1. Подготовить стерильные чашки Петри.
2. Подготовить чистый сосуд, в котором будет проводиться приготовление среды.
3. Сварить мясной бульон.
4. В чистый сосуд насыпать агар в конечной концентрации около 1%.
5. Добавить сахар в конечной концентрации около 3%.
6. Налить в этот же сосуд мясной бульон.
7. Полученную смесь, помешивая греть на водяной бане, до появления маленьких пузырьков и загустевания смеси.
8. Разлить полученную жидкость в чашки Петри, закрыть, после полного загустения среды можно проводить посев.

1.2.2. Посев.

1. Закрытую чашку Петри, с готовой стерильной средой доставить на место посева.
2. Открыть чашку, отойти от неё, не создавать вокруг неё лишних воздушных потоков.
3. Подождать десять минут.
4. Подойти к чашке Петри и аккуратно закрыть её крышкой.
5. Подписать место взятия пробы.
6. Заклеить чашку парафильмом.
7. Поставить чашку Петри в тёмное и тёплое место и оставить её там на 1-3 часа.

1.2.3. Подсчёт колоний бактерий и грибов.

Подсчёт бактерий производится через 1-3 часа после посева, грибов – через 5 – 7 часов. Для подсчёта потребуется защитная маска (многие из выросших колоний могут оказаться болезнетворными), специальный фломастер (для того, чтобы отмечать подсчитанные колонии), бумага для записи. Чашку Петри осматривают со всех сторон, подсчитывая колонии, отмечая их фломастером и характеризуя их. Обычно чашку не открывают, но для более подробного ознакомления с содержимым, надев маску, её можно открыть.

План характеристики колоний:

- А) Форма (круглая, амёбовидная...)
- Б) Окраска (бесцветная, зелёная...)
- В) Поверхность и профиль колоний (гладкая, выпуклая...)
- Г) Блеск и прозрачность (блестящая, прозрачная...)

Д) Размер колонии (в диаметре), измеряется в мм.

По размеру колонии различаются на:

1. Точечные (менее 1 мм в диаметре)
2. Мелкие (1 – 2 мм в диаметре)
3. Средние (2 – 4 мм в диаметре)
4. Крупные (более 4 мм в диаметре)

Е) Край колонии (ровный, волнистый), определяют при малом увеличении микроскопа.

Ж) Структура колонии (однородная, мелкозернистая...), определяют так же при малом увеличении микроскопа.

Если на питательной среде выросло много маленьких колоний, то их примерное число определяют, подсчитав их количество в небольшом секторе и подставив это значение в формулу: $k_1/k_2 = s_1/s_2$, где k_1 и k_2 – это количество колоний в небольшом секторе и на всей питательной среде (величина, которую надо найти), а s_1 и s_2 – это площадь поверхности сектора и всей питательной среды (данные величины измеряются с помощью линейки).

1.2.4. Полученные при работе данные.

Мы проводили посевы бактерий в помещениях: в первой чашке, в 8, 9, 34. Данные, полученные нами в результате посева приводятся внизу в виде таблицы. Посев проводился по алгоритму описанному в пункте №2.

Таблица 1.2.1 -Данные о колониях бактерий, выросших на среде «мясной агар».

свойства	1 чашка	8 чашка	9 чашка	34 чашка	2 чашка	3 чашка
Кол-во	1+1+2	3	1+1+1+1 +1+1	5+1+1+2	1+1+1+1	1
Форма	Снежинки	Снежинка	3 круглых 3 снежинки	5.неправильных 4 круглых	3 круглых 1 неправильная	Амёбовидная
Окраска	2 бледно-белые 1 жёлтая	Белая	1 жёлтая 1 оранжевая 4 белых	3 жёлтые 1 белая 5 бледно-белых	2 белых 1 жёлтая 1 оранжевая	Бледно-жёлтая
Поверхность	2 гладкие 1 рыхлая	Рыхлая	Гладкая	5 рыхлых 4 гладких	Гладкая	Рыхлая
Профиль	2 кратера 1 разная	Выпуклые	3 выпуклых 3 вогнутых	4 выпуклые 5 разных	Выпуклый	Разный
Блеск	Матовые	Матовые	2 блестящие 4 матовые	3 блестящих 6 матовых	Блестящие	Матовая
Прозрачность	Непрозрачные	Непрозрачные	4 непрозрачные 2 прозрачные	5.прозрачных 4 непрозрачных	Непрозрачные	Прозрачная
Размер	1 огромная 2 средних	Маленькие	5 средних 1 крупная	5 средних 1 большая 3 маленькие	Большие	Средняя
Край	1 изрезанный 2 ровный	Изрезанный	Ровный	5 неровный 4 ровный	Ровный	Изрезанный
Структура	2 однородные	Однородная	4 однородные	5 неоднородных	Однородные	Зернистая

	1 зернистая		2 неоднор одные	4 однородные		
Свойства	8 чашка	9 чашка	34 чашка	1 чашка	2 чашка	3 чашка
Кол-во	2+1+1	1+1	5+1+1+2	3	2+5	0
Форма	2 снежинки 2 круглые	1 круглая 1 снежинка	4 круглые 5 неровных	Круглые	6 круглых 1 по краю чашки	-
Окраска	3 белые 1 оранжевая	1 белая 1 жёлтая	1 белая 1 жёлтая 5 бледно- белых 2 бледно- жёлтые	Белые	2 жёлтые 5 белых	-
Поверхнос ть	2 шероховат ые 2 ровные	-	шерохова тая	Ровная	Шероховат ые	-
Профиль	Выпуклые	-	4 выпуклые 5 ровные	Выпуклы й	Разный	-
Блеск	3 матовые 1 блестящая	Матовые	Матовые	Матовые	Матовые	-
Прозрачно сть	Непрозрачн ые	1 прозрачная 1 непрозрачная	2 непрозра чные 7 прозрачн ых	Прозрачн ые	Непрозрачн ые	-
Размер	Маленькие	Маленькие	5маленьк их 4 средних	Маленьки е	1 огромная 2 средних 4 маленьких	-
Край	3 ровных 1 неровный	-	5 неровных 4 ровных	Ровный	1 неровный 6 ровный	-
Структура	1 ровная 3 зернистых	-	5 зернисты х 4 ровных	Зернистая	1 зернистая 6 ровных	-

Таблица 1.2.2- Среда «мясной агар», бактерии, 8, 9, 34 чашка, 1, 2, 3 чашка после уроков.

На среде «мясной агар» не выросло грибов.

2. Данные о колониях бактерий и грибов, выросших на среде Чапека.

Таблица 1.2.3. Среда Чапека. Грибы и бактерии.

Место	До первой проверки		После второй проверки	
	Грибы	Бактерии	Грибы	Бактерии
8 чашка	7 маленьких, зелёных грибов	0	Большой, белый гриб	0
9 чашка	1 гриб	4 маленьких колонии	0	0
34 чашка		0	0	0
1 чашка	17 сизых маленьких грибов, 50 белых мелких грибов	1 средняя колония	0	0
2 чашка	0	0	0	0
3 чашка	0	0	12 больших, зелёных грибов	0

1.2.5. Анализ.

По санитарным нормам воздух считается чистым, если в нём содержится меньше 2500 микроорганизмов на 1м³.

Для определения микробного числа подсчитывают колонии выросшие на чашках Петри (площадь поверхности агара в чашке равна 75 см²) и расчет ведут по правилу В.Л. Омелянского: на поверхность площадью 100 см² за 5 мин оседает такое количество микробов, которое содержится в 10 л воздуха. Расчёт проводится по формуле:

$$X = \frac{A \times 100 \times 100}{75 \text{ см}^2}$$

Где, X - количество микробов в 1 м³; A - количество колоний на питательной среде в чашке Петри.

Таблица 1.2.4. Критерии оценки воздуха жилых помещений

Оценка воздуха	Общее количество бактерий в 1 м ³
Лето чистый загрязненный	до 1500 до 2500
Зима чистый загрязненный	до 4500 до 7000

1.2.6. Выводы.

Мы, проведя подсчёты, пришли к выводу, что все места в которых мы проводили посевы пригодны для использования и соответствуют санитарным нормам.

Таблица 1.2.5 Результаты

Место	Количество бактерий в 1 м ³
До первой проверки	1 чашка 533
	2 чашка 533
	3 чашка 133
	8 чашка 400
	9 чашка 800
	34 чашка 1200
После второй проверки	1 чашка 400
	2 чашка 933
	3 чашка Близится к нулю
	8 чашка 533
	9 чашка 266
	34 чашка 1200

Некоторые значения показались нам слишком идеальными и поэтому, проанализировав все обстоятельства, мы выдвинули гипотезу о том, что в некоторых местах количество бактерий на 1 м³ столь мало из-за того, что за несколько дней до посева была произведена дезинфекция помещений в связи с эпидемией гриппа А(Н1N1). Если гипотеза верна – можно сделать вывод о том, что дезинфекция была произведена качественно.

В начале наших исследований мы предположили, что в 34 чашке количество бактерий будет резко отличаться от количества бактерий в других чашках, из-за

специфичной атмосферы, образованной в результате проведения химических опытов. Это предположение подтвердилось – в этой чашке самый высокий показатель загрязнённости воздуха микроорганизмами.

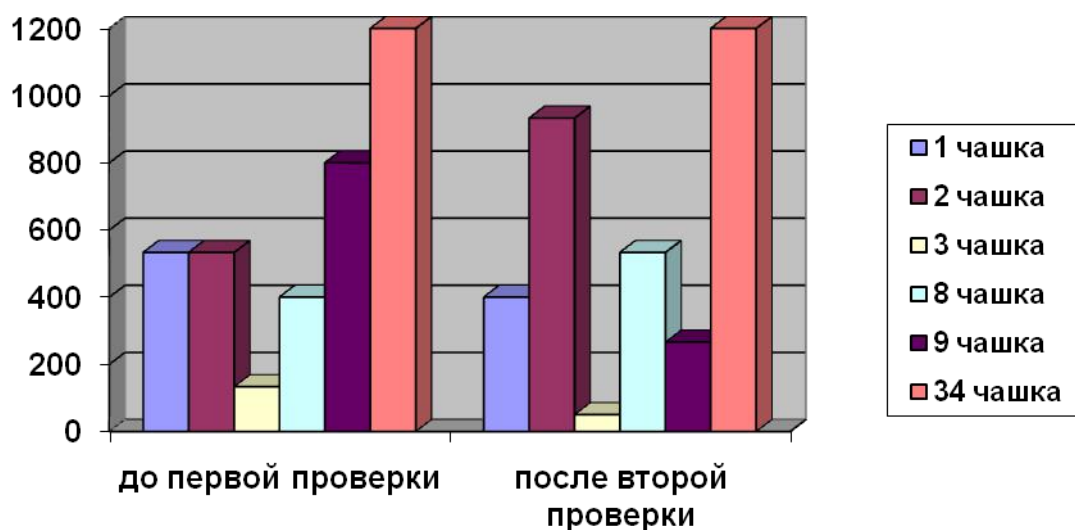


Рисунок 1.2.1. Загрязнённость воздуха

На данной диаграмме соотнесены значения количества бактерий в помещениях. Некоторые значения выросли (2 чашка, 8 чашка), это произошло потому, что рабочие принесли с собой новых бактерий и заодно подняли с пола осевших за ночь микроорганизмов. Некоторые остались неизменными (34 чашка), это может быть объяснено редкой посещаемостью данного места учениками или проветриванием этого места. Но некоторые (1, 3 чашки, 9 чашка) уменьшились. Это наверняка произошло потому что данные помещения сразу после проверки проветривались, и бактерии находящиеся в воздухе улетели на улицу.

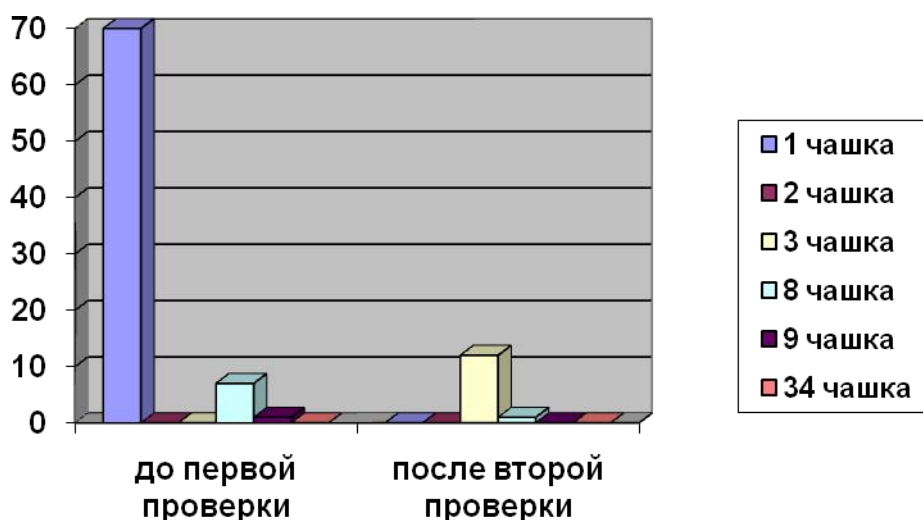


Рисунок 1.2.2. Соотнесение количества грибов.

На данной диаграмме соотнесены значения количества плесневых грибов в помещениях. Самое заражённое грибами это 1 чашка. Присутствие в это помещении до очистки может вызвать аллергию и другие неприятные последствия.

1.3. Ювенильные микроколонии, особенности распознавания.

Особенностью нашей работы является то, что колония должна вырасти за ограниченное время. Это время меньше, чем время, отводимое в обычных больницах. Из-за этого факта возникает проблема - мы не можем заняться распознаванием колоний как в обычном случае, так как они меньше и их структуру сложнее разобрать. Ниже приведены признаки по которым будет проводиться распознавание:

Морфологические свойства по следующим критериям:

1.2.1. диаметр колоний:

- крупные (диаметр более 5 мм)
- средние (диаметр 2-4 мм)
- мелкие (диаметр 1-2 мм)
- карликовые (менее 1 мм)

1.2.2. форма профиля колоний:

- правильная
- неправильная

- круглая
- сферическая
- плоская
- выпуклые
- ворсинчатые
- игольчатые
- зубчатые
- шиповидные
- кратерообразные
- конусовидные

1.2.3. поверхность колоний:

- гладкая
- шероховатая
- бугристая
- бороздчатая
- складчатая
- морщинистая

1.2.4. форма края колоний:

- ровный
- неровный
- волнистый
- фестончатый
- лопастной
- рваный
- зазубренный
- опушенный

1.2.5. консистенция колоний:

- сметанообразная
- крошковатая
- комковатая

- пленчатая
- сухая
- влажная
- слизистая
- тягучая
- плотная
- рыхлая
- вырастающая в среду

1.2.6. прозрачность колоний:

- прозрачные
- полупрозрачные
- непрозрачные

1.2.7. цвет колоний ¹:

- неокрашенные
- белые
- палевые
- беловатые
- кремовые
- бледно-серые
- темно-песочные
- буровато-желтые

1.2.8. структура колоний:

- гомогенная
- зернистая
- однородная
- разнородная в центре и по периферии

1.2.9. характер блеска колоний:

- матовый

¹ Цвет колоний на питательной среде (такой-то) после инкубации при температуре (такой-то) в течение (столько-то и более суток).

- глянцевый
- блестящий

1.2.10. форма микробных клеток:

- кокки
- палочки (прямые, изогнутые)
- коккобациллы

1.2.11. размер микробных клеток:

- мелкие (длина менее 1 мкм),
- средние (длина 1-2 мкм)
- крупные (длина более 2 мкм)

Можно сделать вывод, что в работе нам требуется микроскоп с увеличением $\times 100$, а также программное обеспечение, написанное для него.

1.4 Различные методы распознавания образов. Метод шаблона. Пространство признаков.

С задачей распознавания образов живые системы, в том числе и человек, сталкиваются постоянно с момента своего появления. В частности, информация, поступающая с органов чувств, обрабатывается мозгом, который в свою очередь сортирует информацию, обеспечивает принятие решения, а далее с помощью электрохимических импульсов передает необходимый сигнал далее, например, органам движения, которые реализуют необходимые действия. Затем происходит изменение окружающей обстановки, и вышеуказанные явления происходят заново. И если разобраться, то каждый этап сопровождается распознаванием.

С развитием вычислительной техники стало возможным решить ряд задач, возникающих в процессе жизнедеятельности, облегчить, ускорить, повысить качество результата. К примеру, работа различных систем жизнеобеспечения, взаимодействие человека с компьютером, появление роботизированных систем и др. Тем не менее, отметим, что обеспечить удовлетворительный результат в некоторых задачах (распознавание быстро движущихся подобных объектов, рукописного текста) в настоящее время не удастся.

Уже сейчас распознавание образов плотно вошло в повседневную жизнь и является одним из самых насущных знаний современного инженера. В медицине распознавание образов помогает врачам ставить более точные диагнозы, на заводах оно используется для прогноза брака в партиях товаров. Системы биометрической идентификации личности в качестве своего алгоритмического ядра так же основаны на результатах этой дисциплины. Дальнейшее развитие искусственного интеллекта, в частности проектирование компьютеров пятого поколения, способных к более непосредственному общению с человеком на естественных для людей языках и посредством речи, немыслимы без распознавания. Здесь рукой подать и до робототехники, искусственных систем управления, содержащих в качестве жизненно важных подсистем системы распознавания .

- Примеры задач распознавания образов:
- Распознавание букв;
 - Распознавание штрих-кодов;
 - Распознавание автомобильных номеров;
 - Распознавание лиц и других биометрических данных;
 - Распознавание речи

В целом, можно выделить три метода распознавания образов: Метод перебора. В этом случае производится сравнение с базой данных, где для каждого вида объектов представлены всевозможные модификации отображения. Например, для оптического распознавания образов можно применить метод перебора вида объекта под различными углами, масштабами, смещениями, деформациями и т. д. Для букв нужно перебирать шрифт, свойства шрифта и т. д. В случае распознавания звуковых образов, соответственно, происходит сравнение с некоторыми известными шаблонами (например, слово, произнесенное несколькими людьми).

Второй подход - производится более глубокий анализ характеристик образа. В случае оптического распознавания это может быть определение различных геометрических характеристик. Звуковой образец в этом случае подвергается частотному, амплитудному анализу и т. д.

Следующий метод - использование искусственных нейронных сетей (ИНС). Этот метод требует либо большого количества примеров задачи распознавания при обучении, либо специальной структуры нейронной сети, учитывающей специфику данной задачи. Тем не менее, его отличает более высокая эффективность и производительность.

Объект в распознавании образов описывается совокупностью основных характеристик (признаков, свойств). Основные характеристики могут иметь различную природу: они могут браться из упорядоченного множества типа вещественной прямой, либо из дискретного множества (которое, впрочем, так же может быть наделено структурой). Такое понимание объекта согласуется как потребностью практических приложений распознавания образов, так и с нашим пониманием механизма восприятия объекта человеком. Действительно, мы полагаем, что при наблюдении (измерении) объекта человеком, сведения о нем поступают по конечному числу сенсоров (анализируемых каналов) в мозг, и каждому сенсору можно сопоставить соответствующую характеристику объекта. Помимо признаков, соответствующих нашим измерениям объекта, существует так же выделенный признак, либо группа признаков, которые мы называем классифицирующими признаками, и в выяснении их значений при заданном векторе X и состоит задача, которую выполняют естественные и искусственные распознающие системы.

Понятно, что для того, чтобы установить значения этих признаков, необходимо иметь информацию о том, как связаны известные признаки с классифицирующими. Информация об этой связи задается в форме прецедентов, то есть множества описаний объектов с известными значениями классифицирующих признаков. И по этой прецедентной информации и требуется построить решающее правило, которое будет ставить произвольному описанию объекта значения его классифицирующих признаков.

Для идентификации таких сложных объектов, как микробиологические объекты (бактерии, колонии бактерий) представляется целесообразным

использовать алгоритмы распознавания, основанные на концепции пространства признаков.

Дело в том, что в отличие от объектов с заданной топологией, таких как букв, например, шаблонный метод или метод наложения не дает удовлетворительных результатов. Это происходит из-за того, что представители одного и того же вида бактерий, могут существенно отличаться по форме, сохраняя, тем не менее, некоторые инвариантные свойства или признаки, по которым их, например, отличает визуально лаборант, при рассмотрении в микроскопе.

Каждый признак представляет собой вещественное число и может быть отложен по оси координат. Количество признаков, таким образом, определяет размерность пространства признаков. Как сами признаки, как и размерность должны быть выбраны оптимальным образом для того, чтобы метод обладал специфичностью, т.е. мог бы с уверенностью отличать один вид микроскопических биообъектов от других. Объект идентифицируется при попадании его в заранее определенную в алгоритме область пространства признаков. Такой областью обычно бывает эллипсоид (в 3-х или более измерениях). На рисунке 1.1 представлен пример, иллюстрирующий ситуацию, когда в двумерной проекции (рисунок 1.6 а) области могут прикрываться и идентификация объектов становится невозможной. При введении третьего признака (третьей координаты), области становятся пространственно разнесенными, и идентификация не представляет труда.

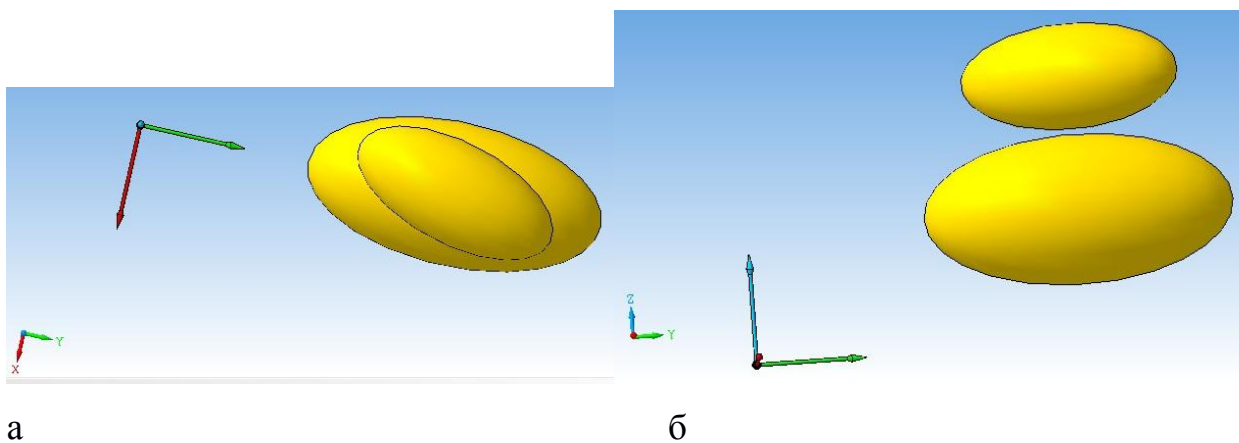


Рисунок 1.4.11 Пример пространства признаков: а) пересечения областей в проекции б) непересекающиеся области

1.5 Сравнение современных программных реализаций анализа изображений для биомедицинских приложений

В данном разделе мы рассмотрим основные программы для анализа изображений в биомедицине.

- Colonyzer – автоматизированная система квантификации роста микроорганизмов на твердой агар-агаровой подложке
- CDSS (Clinical Decision Support System – "Система поддержки принятия клинических решений") классифицирует изображения на основе экспертных знаний с целью помощи врачам в принятии ими решений. Грид-инфраструктура EGEE используется как для набора больших объёмов данных, так и для эффективного "обучения" классифицирующего программного обеспечения на этих данных.
- Pharmacokinetics – изучение диффузии контрастных агентов в печени по последовательности изображений, полученных с помощью магнитного резонанса. Артефакты, связанные с движениями пациента, не позволяют напрямую сравнивать изображения. В грид-инфраструктуре, однако, можно за разумное время выполнить распараллеленный анализ последовательности изображений.
- SiMRI3D – моделирование на основе изображений, полученных с помощью магнитного резонанса (МР). Цель моделирования – получить

искусственные, но реалистичные трёхмерные МР-изображения для анализа изображений от хорошо известных объектов, изучения артефактов и дальнейшего развития и совершенствования технологии МР-изображений.

■ **gPTM3D** – интерактивное восстановление трёхмерных медицинских изображений: например, изображения всего объёма сложных органов. Требования к качеству интерактивной работы сервисов таковы, что некоторые сайты в грид-инфраструктуре должны установить высокий приоритет для таких задач.

■ **Bronze Standard** – оценка алгоритмов получения медицинских изображений. Объём данных и стоимость вычислений недоступны обычным компьютерам, но в грид-инфраструктуре это приложение работает довольно легко.

Пакет программного обеспечения SPM применяется в нейрологических исследованиях для ранней диагностики болезни Альцгеймера. В его основе лежит сравнение данных пациента с большим набором данных от людей без этой патологии. Грид-технологии предоставляют лёгкий доступ к распределённым данным и распределённым вычислительным ресурсам.

1.6 Open CV для Android. Обзор.

OpenCV — это библиотека алгоритмов компьютерного зрения, обработки изображений. Исходный код является открытым.

Эта вещь очень интересная и ей можно найти много применений: идентификация пользователя за компьютером, распознавание предметов, трэкинг объектов, а уж в робототехнике применений ещё больше! С 2000 по 2008 год OpenCV разрабатывалась и поддерживалась в основном Intel, и Нижегородское отделение корпорации с самого начала играло ведущую роль в формировании облика библиотеки. В первые годы существования OpenCV стремительно расширялась вширь, обрастая базовой функциональностью, такой как основные структуры данных, алгоритмы обработки изображений (image processing), базовые алгоритмы компьютерного зрения, ввод и вывод изображений и видео. Уже в это время были реализованы алгоритмы детекции человеческих лиц (каскадный

классификатор), поиска стереосоответствия, оптического потока и другие. Однако, в 2004 году Intel практически прекратил поддержку библиотеки. Многие участники проекта, которые к тому времени уже сложились как эксперты в области компьютерного зрения, не захотели переквалифицироваться, и покинули компанию. Фактически, именно так возник Itseez, деятельность которого в значительной степени базируется на опыте, полученном во время разработки OpenCV.

Второй мощный толчок к развитию проект получил с приходом компании Willow Garage, основной целью которой является создание персонального робота. OpenCV стала важной частью ROS (Robot Operating System), и на ее основе был создан ряд компонент для робота PR2, но об этом мы расскажем как-нибудь позже. Начало этого периода совпадает с переходом в Itseez из Intel Вадима Писаревского (aka OpenCV Tsar) в 2008 году. При поддержке Willow была сформирована команда, стартовавшая существенную переработку библиотеки. Именно следствием этих усилий стало то, что OpenCV приобрела C++ и Python API, модуль `features2d`, новую архитектуру, билд-систему на основе CMake, систему непрерывной интеграции на основе BuildBot, улучшенную документацию, прекрасные tutorиалы и массу других новшеств. На сегодняшний день можно говорить уже о третьем значительном этапе в жизни библиотеки, наступившем с приходом NVidia. В 2010 году эта компания поддержала создание CUDA-оптимизированной версии библиотеки, что также было сделано инженерами компании Itseez. Первым публичным результатом стала реализация алгоритма стереосоответствия, способная в реальном времени производить обработку видео в FullHD-разрешении (1920x1080 пикселей). На сегодняшний день `opencv_gpu` — это полновесный модуль, нашедший применение во многих промышленных приложениях. Начиная с 2011 года список задач расширился, и Itseez совместно с NVidia начал разработку версии библиотеки под Android, с оптимизациями для платформы NVidia Tegra. OpenCV4Android пережила уже несколько релизов, и приобрела свое собственное сообщество пользователей. Кроме того, OpenCV обогатилась

Java API (пока только для Android), встроенной системой тестирования производительности и рядом улучшений.

1.7 Постановка задачи дипломного проекта.

Дано множество объектов и входное цветное изображение, полученное с помощью камеры мобильного телефона, либо обычной камеры на микроскопе. Относительно большого числа объектов необходимо провести классификацию объекта на входном изображении. Требуется по имеющейся информации о объектах и изображении объекта установить к какому виду бактерии относится объект на изображении.

На входном изображении объекта необходимо сначала произвести цветовую фильтрацию изображения для устранения помех и шумов, а затем провести сегментацию и выделение контуров объектов, затем провести расчёты и классифицировать объект в пространстве признаков.

Задача идентификации, которая заключается в том, чтобы выделить определенный конкретный объект среди ему подобных (например, узнать среди других людей свою жену).

Отнесения объекта к тому или иному классу. Это может быть, например, задача распознавания букв или принятия решения о наличии дефекта в некоторой технической детали. Отнесения объекта к определенному классу отражает наиболее типичную проблему классификации, и, когда говорят о распознавании образов, чаще всего подразумевают именно эту проблему. Именно она рассматривается здесь в первую очередь.

Кластерный анализ, заключающийся в разделении заданного набора объектов на классы - группы объектов, схожи между собой по тем или иным критерием. Эту задачу часто называют классификацией без учителя, поскольку, в отличие от задачи 2, классы априорно не заданы.

Проблемы распознавания легко решаются людьми, причем делается это, как правило, подсознательно. Попытки же построить искусственные системы распознавания не столь убедительны. Основная проблема заключается в том, что часто невозможно адекватно определить признаки, на основе которых следует

осуществлять распознавание. Для задач, для которых такие признаки удастся выделить, искусственные системы распознавания получили значительное распространение и широко используются.

Формальная постановка задачи

Распознавание образов - это отнесение исходных данных к определенному классу с помощью выделения существенных признаков, характеризующих эти данные, из общей массы несущественных данных.

При постановке задач распознавания пытаются пользоваться математическим языком, стараясь, в отличие от теории искусственных нейронных сетей, где основой является получение результата путем эксперимента, заменить эксперимент логическими рассуждениями и математическими доказательствами.

Зачастую в задачах распознавания образов рассматриваются монохромные изображения, что позволяет рассматривать изображение как функцию на плоскости. Если рассмотреть множество точек на плоскости T , где функция $x(x, y)$ выражает в каждой точке изображения его характеристику - яркость, прозрачность, оптическую плотность, то такая функция является формальной записью изображения.

Множество же всех возможных функций $x(x, y)$ на плоскости T - является моделью множества всех изображений X . Вводя понятие сходства между образами можно поставить задачи распознавания. Конкретный вид такой постановки сильно зависит от следующих этапов при распознавании согласно того или иного подхода.

Глава 2 Разработка программного обеспечения для системы распознавания образов.

2.1 Получение базы данных контрастных изображений колоний

База контуров объектов используется для нахождения на входном изображении объекта и его ракурса. По некоторому набору моделей генерируются их изображения в различных ракурсах и в разных масштабах. Затем на этих изображениях детектируются контуры и выделяются особые точки. Эти результаты можно сохранить для дальнейшего переиспользования. Так же можно поделить эту базу на группы в зависимости от размеров контуров в пикселях. Также мы составляем базу в соответствии с результатами выделения цветов и их отображения на графике пространства признаков.

Возможности программы:

- хранение произвольного количества изображений, фотографий, видеофайлов и документов в одной записи
- хранение неограниченного количества текстовой и цифровой информации в записи
- развитые средства импорта изображений из цифровой камеры и из каталогов
- автоматическое разделение всех импортируемых изображений на отдельные записи по сериям
- быстрый поиск записи по любому текстовому полю, по фамилии пациента достаточно ввести 1-3 первые буквы
- легкое формирование фильтров с наборами критериев для поиска по нескольким условиям
- оперативное включение различных фильтров
- удобные средства просмотра
- легкая модификация набора полей базы в визуальном режиме формирования бланка
- инструмент для быстрого заполнения полей готовыми

блоками текста

- хранение в записи файлов любых типов, *.doc, *.xls и т.д.
- возможность легко изменять и обрабатывать изображения в других программах, продолжая хранить их в базе данных
- настраиваемый отчет для вывода данных на печать, возможность изменения шаблонов печати для вывода данных в различных формах
- возможность использования одной базы изображений в сети несколькими пользователями
- ввод изображений с любых камер, сканеров и других устройств, поддерживающих TWAIN или DirectShow протоколы
- линейные и угловые измерения, измерения кругов: радиус, диаметр, площадь и объем шара
- измерения на живом видео и на изображениях
- печать отчетов

2.2. Цифровая фильтрация изображений

Задачей обработки изображения может быть как улучшение (восстановление, реставрация) изображения по какому-то определенному критерию, так и специальное преобразование, кардинально меняющее изображение. В последнем случае обработка изображений может быть промежуточным этапом для дальнейшего распознавания изображения (например, для выделения контура объекта). Методы обработки изображения могут существенно различаться в зависимости от того, каким путем изображение было получено — синтезировано системой машинной графики, либо путем оцифровки черно-белой или цветной фотографии или видео. В том случае, если изображение было получено с помощью оцифровки, на них, как правило, присутствует шум.

Обычно изображения получаемые из каких-либо источников искажаются действием помех. Это затрудняет их распознавание как человеком, так и персональным компьютером либо иным средством обработки. Ослабление действия помех осуществляется при помощи фильтрации. Изображение представляет собой двумерную матрицу, где каждой точке соответствует пиксель, обладающий такими параметрами как яркость и интенсивность красного, синего и зелёного цветов. Идеей фильтрации является использование как свойств точки, так и её окрестностей. Наша задача состоит в том, чтобы найти оптимальный алгоритм для автоматической фильтрации изображений, путём избавления от помех. Выделяют частотную и пространственную фильтрацию.

Частотные методы используют в своей основе преобразование Фурье, путём разложения изображения на тригонометрические функции. Затем путём добавления значений к этим функциям, мы можем избавиться от шума. Пространственный метод осуществляется путём рассмотрения изображения как двумерной матрицы и применения к ней различных операторов, таким образом изменяя значения пикселей. Матрицы изменения бывают разными – как квадратными так и прямоугольными и называются масками. Пространственные методы делятся на линейные и нелинейные. Разница в том, что в линейном методе матрица делается согласно формуле:

$$g(x, y) = \sum_{s=\frac{(m-1)}{2}}^{\frac{(m-1)}{2}} \sum_{t=\frac{(n-1)}{2}}^{\frac{(n-1)}{2}} w(s, t) f(x + s, y + t) \quad (2.2.1)$$

$x=0,1,2,...,M-1$, $y=0,1,2,...,N-1$ – координаты исходного изображения f . Неудобство данного метода в том, что здесь исключён индивидуальный подход для каждого изображения, что может привести к дополнительным шумам. Поэтому стоит использовать нелинейный метод, когда матрица задаётся человеком самостоятельно, либо используется готовая матрица, на основе уже созданных. Рассмотрим медианную фильтрацию и адаптивную. В этом случае мы считаем средние значения пикселей находящихся в области фильтрации, таким образом

делая сам объект практически одноцветным, но при этом чётко выделяя границы объекта, что способствует распознаванию. Адаптивный фильтр использует алгоритм Винера, сглаживая объекты, сохраняя цвета и границы высокочастотных областей. В большинстве случаев, если нам необходимо знание цвета объекта, то используется именно этот алгоритм, но в нашем случае ввиду малой времени обработки, а также работе с чёрно белыми изображениями мы отдаём предпочтение медианному виду фильтрации.

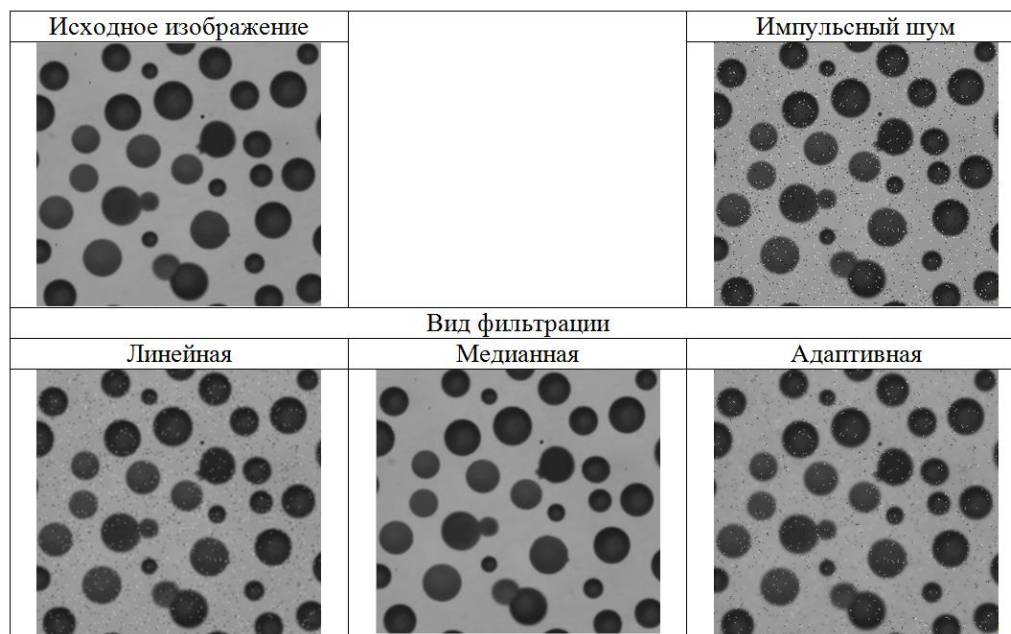


Рисунок. 2.2.1. Результаты фильтрации изображений при обработке импульсного шума.

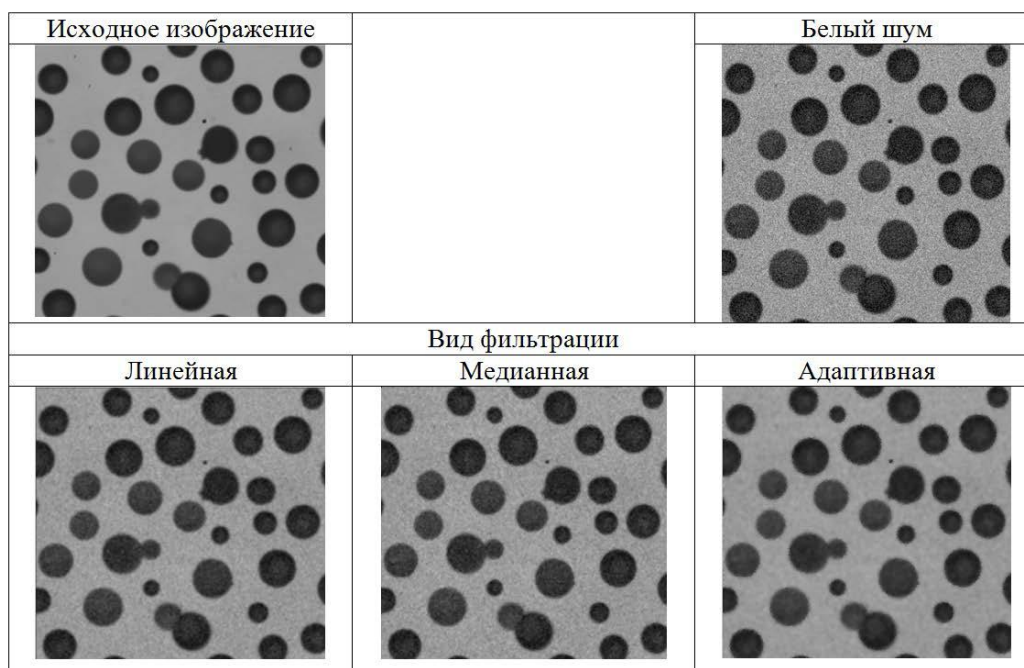


Рисунок. 2.2.2. Результаты фильтрации изображений при обработке белого шума.

В таблице представлены результаты сравнения предлагаемого алгоритма и стандартных.

Таблица 2.2.1 – Скорость работы исследуемых алгоритмов

	wiener2	medfilt2	new LPA+ICI
Гауссов	0,0024	0,0038	0,0016
Импульсный	0,0087	0,0028	0,0022
Комбинированный	0,0094	0,0043	0,0032

Итак, нами были исследованы различные

подходы к удалению трех основных видов шума: Гауссов шум, импульсный шум, комбинированный шум. Был разработан адаптивный метод удаления этих шумов. На его основе были разработан алгоритм, который был реализован в среде Matlab. Была создана система, удаляющая перечисленные шумы эффективнее существующих систем. Кроме того, эта система универсальна и сама подстраивается по имеющийся шум. Был проведен сравнительный анализ работы стандартных систем и новой системы. Было установлено, что во всех случаях новая система работает эффективнее стандартных. На это указывают как объективные данные, такие, как подсчет СКО, так и субъективные, такие, как визуальная оценка результатов работы пользователем.

В дальнейшем планируется адаптировать данные подходы для обработки не только черно-белых, но и цветных изображений. Планируется создание гибридных алгоритмов, совмещающих эффективные составляющие новых и стандартных алгоритмов.

2.3. Выделение контуров колоний.

С точки зрения распознавания и анализа объектов на изображении наиболее информативными являются не значения яркостей объектов, а характеристики их границ – контуров. Другими словами, основная информация

заклучена не в яркости отдельных областей, а в их очертаниях. Задача выделения контуров состоит в построении изображения именно границ объектов и очертаний однородных областей.

Как правило, граница предмета на фотографии отображается перепадом яркости между двумя сравнительно однотонными областями. Но перепад яркости может быть вызван также текстурой предмета, тенями, бликами, перепадами освещенности, и т.п.

Будем называть контуром изображения совокупность его пикселей, в окрестности которых наблюдается скачкообразное изменение функции яркости. Так как при цифровой обработке изображение представлено как функция целочисленных аргументов, то контуры представляются линиями шириной, как минимум, в один пиксел. Если исходное изображение, кроме областей постоянной яркости, содержит участки с плавно меняющейся яркостью, то непрерывность контурных линий не гарантируется. С другой стороны, если на “кусочно-постоянном” изображении присутствует шум, то могут быть обнаружены “лишние” контуры в точках, которые не являются границами областей.

При разработке алгоритмов выделения контуров нужно учитывать указанные особенности поведения контурных линий. Специальная дополнительная обработка выделенных контуров позволяет устранять разрывы и подавлять ложные контурные линии.

Процедура построения бинарного изображения границ объектов обычно складывается из двух последовательных операций: выделения контуров и их пороговой обработки.

Исходное изображение подвергается линейной или нелинейной обработке, с реакцией на перепады яркости. В результате этой операции формируется изображение, функция яркости которого существенно отличается от нуля только в областях резких изменений яркости изображения. Пороговой обработкой из этого изображения формируется контурный объект. Выбор порога на втором этапе должен производиться из следующих соображений. При слишком высоком

пороге могут появиться разрывы контуров, а слабые перепады яркости не будут обнаружены. При слишком низком пороге из-за шумов и неоднородности областей могут появиться ложные контуры.

2.3.1 Поиск границ на основе градиента.

Одним из наиболее простых способов выделения границ является пространственное дифференцирование функции яркости. Для двумерной функции яркости $A(x, y)$ перепады в направлениях x и y регистрируются частными производными $A(x, y)/x$ и $A(x, y)/y$, которые пропорциональны скоростям изменения яркости в соответствующих направлениях.

Выделение перепадов яркости иллюстрирует рисунок 2.3.1. На нем можно видеть, что подчеркивание контуров, перпендикулярных к оси x , обеспечивает производная $A(x, y)/x$ (рис. б), а подчеркивание контуров, перпендикулярных к оси y , – $A(x, y)/y$ (рис. в).

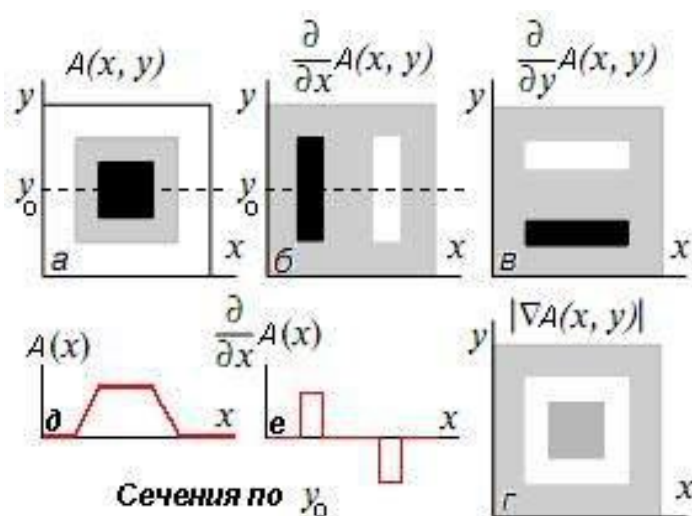


Рисунок 2.3.1. Перепады яркости

В практических задачах требуется выделять контуры, направление которых является произвольным. Для этих целей можно использовать модуль градиента функции яркости

$$|\nabla A(x, y)| = \sqrt{\left(\frac{\partial A(x, y)}{\partial x}\right)^2 + \left(\frac{\partial A(x, y)}{\partial y}\right)^2},$$

который пропорционален максимальной (по направлению) скорости изменения функции яркости в данной точке и не зависит от направления контура. Модуль градиента в отличие от частных производных принимает только неотрицательные значения, поэтому на получающемся изображении (рис. г) точки, соответствующие контурам, имеют повышенный уровень яркости.

Для цифровых изображений аналогами частных производных и модуля градиента являются разностные функции.

Практический пример выделения границ на фотоизображении приведен на рисунок 2.3.2. Исходное изображение (1) является однотонным. На изображении (2) представлен результат вычисления вектора градиента яркости $\nabla A(x, y) = (\partial A / \partial x, \partial A / \partial y)$. Как видно на рисунке, в точках большого перепада яркости градиент имеет большую длину. Отфильтровав пиксели с длиной градиента, большей определенного порога τ , мы получим изображение границ (3).



Рисунок 2.3.2 Градиентный анализ

Недостаток алгоритма - пропуск границы с малыми перепадами яркости и включение в число границ деталей изображения с большими изменениями яркости (шкурка бурундука). При зашумлении изображения карту граничных точек будут загрязнять и просто шум, поскольку не учитывается, что граничные точки соответствуют не просто перепадам яркости, а перепадам яркости между относительно монотонными областями.

Для снижения влияния данного недостатка изображение сначала

подвергают сглаживающей гауссовской фильтрации. При сглаживающей фильтрации мелкие несущественные детали размываются быстрее перепадов между областями. Результат операции можно видеть на изображении (4). Однако при этом четко выраженные границы расплываются в жирные линии.

Градиент яркости в каждой точке характеризуется длиной и направлением. Выше при поиске граничных точек использовалась только длина вектора. Направление градиента - это направление максимального возрастания функции, что позволяет использовать процедуру подавления немаксимумов. При этой процедуре для каждой точки рассматривается отрезок длиной в несколько пикселей, ориентированный по направлению градиента и с центром в рассматриваемом пикселе. Пиксель считается максимальным тогда и только тогда, когда длина градиента в нем максимальна среди всех длин градиентов пикселей отрезка. Граничными можно признать все максимальные пиксели с длинами градиента больше определенного порога. Градиент яркости в каждой точке перпендикулярен границе, поэтому после подавления немаксимумов жирных линий не остается. На каждом перпендикулярном сечении жирной линии останется один пиксель с максимальной длиной градиента.

Перпендикулярность градиента яркости к границе может быть использована для прослеживания границы, начиная с некоторого граничного пикселя. Такое прослеживание используется в гистерезисной фильтрации максимальных пикселей. Идея гистерезисной фильтрации заключается в том, что длинный устойчивый граничный контур, скорее всего, содержит в себе пиксели с особенно большим перепадом яркости, и, начиная с такого пикселя, контур можно проследить, переходя по граничным пикселям с меньшим перепадом яркости.



Рисунок 2.3.3. Выделение объектов

При проведении гистерезисной фильтрации вводят не одно, а два пороговых значения. Меньшее (\checkmark) соответствует минимальной длине градиента, при которой пиксель может быть признан граничным. Большее (\otimes), соответствует минимальной длине градиента, при которой пиксель может инициализировать контур. После того как контур инициализируется в максимальном пикселе P с длиной градиента, большей \otimes , рассматриваются каждый соседний с ним максимальный пиксель Q . Если пиксель Q имеет длину градиента, большую \checkmark , и угол между векторами PQ и $\nabla(P)$ близок к 90° , то P добавляется к контуру, и процесс рекурсивно переходит к Q . Его результат для исходного изображения на рисунке 2.3.2 показан на рисунке 2.3.3.

Таким образом, алгоритм нахождения границ на основе градиента заключается в последовательном применении следующих операций:

- гауссовская сглаживающая фильтрация;
- нахождение градиента яркости в каждом пикселе;
- нахождение максимальных пикселей;
- гистерезисная фильтрация максимальных пикселей.

Этот алгоритм носит названия алгоритма Кэнни и наиболее часто применяется для нахождения границ.

2.3.2. Поиск границ на основе лапласиана.

Известно, что необходимым и достаточным условием экстремального значения первой производной функции в произвольной точке является равенство нулю второй производной в этой точке, причем вторая производная должна иметь разные знаки по разные стороны от точки.

В двумерном варианте аналогом второй производной является лапласиан - скалярный оператор $\nabla^2 f = (\partial^2 f / \partial x^2 + \partial^2 f / \partial y^2)$.



Рисунок 2.3.4 Гауссовское сглаживание

Нахождение границ на изображении с использованием лапласиана может производиться по аналогии с одномерным случаем: граничными признаются точки, в которых лапласиан равен нулю и вокруг которых он имеет разные знаки. Оценка лапласиана при помощи линейной фильтрации также предваряется гауссовской сглаживающей фильтрацией, чтобы снизить чувствительность алгоритма к шуму. Гауссовское сглаживание и поиск лапласиана можно осуществить одновременно, поэтому нахождение границ при помощи такого фильтра производится быстрее, чем при помощи алгоритма Кэнни. Фильтр применяется в системах, где имеет значение и качество результата (обычно

уступает алгоритму Кэнни), и быстродействие. Чтобы уменьшить чувствительность к несущественным деталям, из числа граничных точек также можно исключить те, длина градиента в которых меньше определенного порога.

2.4 Выделение признаков для распознавания микробиологических объектов

Выделение признаков позволяет упростить реализацию распознавания или идентификации объектов. При выборе наиболее информативных признаков необходимо учитывать как свойства самих объектов, так и возможности разрешающей способности первичных формирователей сигнала изображения. Выделение признаков проведем на примере обработки монохромных (однослойных) изображений. В цветных изображениях рассмотренные алгоритмы можно применять к каждому цвету в отдельности.

При обработке предпочтительными являются следующие признаки объектов:

- площадь и периметр изображения объекта;
- размеры вписанных простейших геометрических фигур (окружностей, прямоугольников, треугольников и др.);
- число и взаимное расположение углов;
- моменты инерции изображений объектов.

Важной особенностью большинства геометрических признаков является инвариантность относительно разворота изображения объекта, а путем нормирования геометрических признаков друг относительно друга достигается инвариантность относительно масштаба изображения объекта.

2.4.1 Определение площади и периметра. Площадь изображения объекта вычисляется путём подсчёта числа элементов, относящихся к объекту:

$$A = \sum_x \sum_y S(x, y), \quad S(x, y) = \begin{cases} 1, & S(x, y) \in L \\ 0, & S(x, y) \notin L \end{cases},$$

где L - множество координат массива $S(x, y)$, принадлежащих выделенному объекту.

Периметр изображения объекта P вычисляется после выделения границ

объекта путем суммирования множество контурных точек изображения объекта.

На основе выделенных признаков нормированный признак, инвариантный к масштабу изображения $U = A/P^2$ или $V = P/\sqrt{A}$.

2.4.2. Определение радиусов вписанных и описанных окружностей

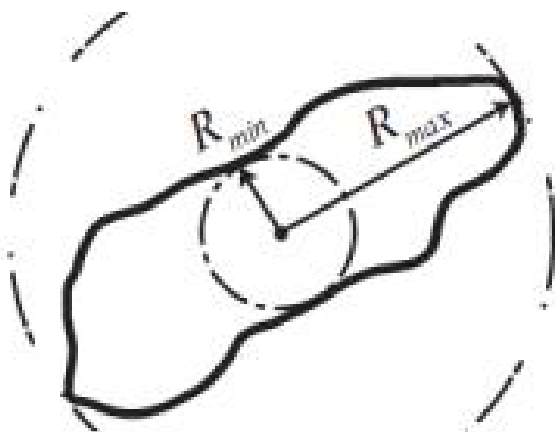


Рисунок 2.4.1 Определение координат центра

Определение радиусов складывается из двух этапов.

1. Определение координат геометрического центра изображения объекта:

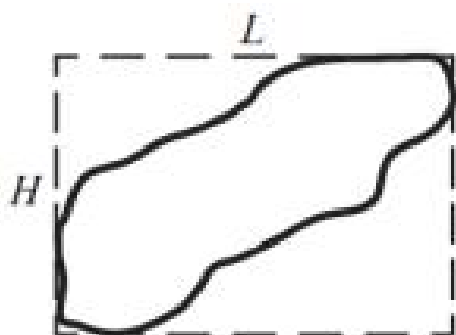
$$X_{ц} = \sum_x \sum_y x S(x, y) / \sum_x \sum_y S(x, y), \quad Y_{ц} = \sum_x \sum_y y S(x, y) / \sum_x \sum_y S(x, y) \quad (2.4.1)$$

где x и y – номера строк и столбцов всех пикселей $S(x, y)$, входящих в объект.

2. Вычисление минимального и максимального расстояний от центра до границ изображения объекта.

$$r(x, y) = \sqrt{(x - X_{ц})^2 + (y - Y_{ц})^2} \quad (2.4.2)$$

$R_{max} = r(x, y_{max}); R_{min} = r(x, y_{min})$, где $x, y \in P$ (точки периметра).



Нормированный признак $R' = R_{max} / R_{min}$ инвариантен к масштабу изображения объекта.

2.4.3. Определение сторон описанного прямоугольника

Рисунок 2.4.2. Определение длин сторон

выполняется следующим образом. Определяются максимальные и минимальные значения абсцисс и ординат изображения объекта x_{\max} и x_{\min} , y_{\max} и y_{\min} , после чего определяются высота и основание прямоугольника:

$$L = x_{\max} - x_{\min}, \quad H = y_{\max} - y_{\min}. \quad (2.4.3)$$

Данный признак не инвариантен к развороту изображения объекта.

2.4.4. Определение числа и взаимного положения углов.

Классический способ определения угловых точек изображения объекта заключается в анализе небольшого фрагмента контура в окрестностях данной точки и в определении радиуса её кривизны. Если радиус окажется меньше установленного порога – это угловой элемент, в противном случае – нет.

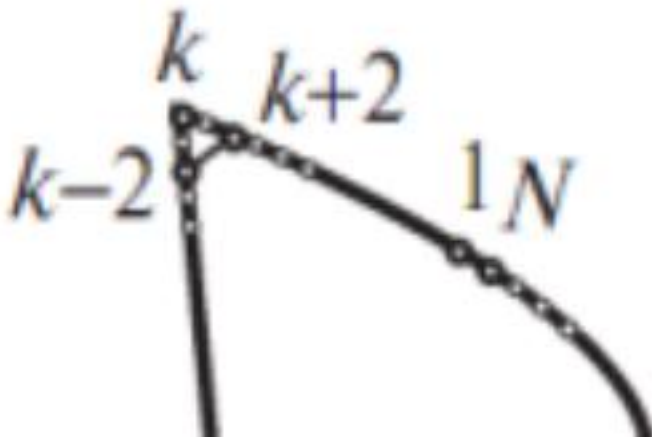


Рисунок 2.4.3 Определение углов

С практической точки зрения предпочтительным представляется более простой алгоритм. Он заключается в оценке расстояний между начальной и конечной точками фрагмента контура, например, между элементами

контура с порядковыми номерами $k - 2$ и $k + 2$

Пусть $x(k)$ и $y(k)$ абсцисса и ордината контурных элементов соответственно. Тогда решающее правило может выглядеть следующим образом:

$$\{|x(k-2)-x(k+2)|+|y(k-2)-y(k+2)| \leq N\}.$$

Если условие выполняется, тогда данная точка контура принадлежит множеству угловых точек L . Здесь N – пороговое значение, выбираемое с учётом свойств объектов.

2.4.5. Определение моментов инерции объекта. Термин "моменты инерции изображения объекта" не имеет отношения к механике. Просто для вычисления указанного признака используются математические выражения, аналогичные вычислению моментов инерции материального тела, где вместо значений масс отдельных точек тела подставлены значения освещенностей в соответствующих точках его изображения. Моменты инерции являются

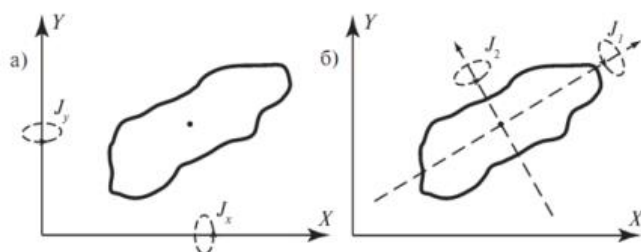


Рисунок 2.4.3 Моменты инерции

информационными признаками для последующего распознавания образов.

Обозначим главные искомые моменты инерции изображения объекта через J_1 и J_2 . Чтобы найти J_1 и J_2 , необходимо предварительно определить так называемые промежуточные моменты J_x и J_y , т.е. моменты инерции относительно вертикальной и горизонтальной осей системы координат, а также смешанный момент $J_{x,y}$.

Порядок вычислений:

1. Определяются координаты центра "тяжести" (энергетического центра) изображения объекта.

$$X_{цз} = \sum_x \sum_y xA(x, y) / \sum_x \sum_y A(x, y), \quad Y_{цз} = \sum_x \sum_y yA(x, y) / \sum_x \sum_y A(x, y) \quad (2.4.4)$$

2. Определяются промежуточные моменты J_x , J_y , $J_{x,y}$.

$$J_x = \sum_x \sum_y [(x - X_{цз})^2 A(x, y)], \quad J_y = \sum_x \sum_y [(y - Y_{цз})^2 A(x, y)] \quad (2.4.5)$$

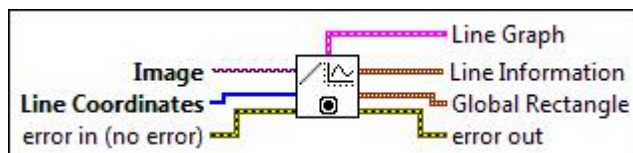
$$J_{xy} = \frac{1}{N} \sum_x \sum_y [(x - X_{\text{цп}})(y - Y_{\text{цп}}) A(x, y)] \quad (2.4.6)$$

3. Вычисляются главные моменты.


$$J_{1,2} = \frac{J_x J_y}{2} \pm \sqrt{\frac{1}{4}(J_x - J_y)^2 + J_{xy}^2} \quad (2.4.7)$$

Наравне с геометрическими признаками мы использовали цветовые идентификаторы. Особенную привлекательность этой идее на ранних этапах развития обработки изображений придавало то, что анализ двумерных полей был сравнительно новой и неизведанной задачей, в то время как в области обработки одномерных функций и сигналов у математиков и специалистов в других областях знания был уже накоплен значительный опыт. Следовательно, выделив из изображения какой-либо характеризующий его одномерный сигнал, можно было применить для его обработки и анализа большое количество уже существующих готовых инструментов. Кроме того, как уже говорилось, обработка одномерных сигналов требует существенно меньшей затраты вычислительных ресурсов, что было крайне важно в те времена, когда вычислительная техника еще не была так мощна и совершенна, как сегодня. Да и сегодня актуальность экономии вычислительных ресурсов сохраняется во всех тех задачах, которые должны решаться системами машинного зрения в реальном масштабе времени. Итак, яркостная составляющая изображения действительно характеризуется одномерным массивом гистограммы (частоты встречаемости на изображениях пикселей одинаковой яркости). Однако геометрические характеристики изображения ($\{x, y\}$) по-прежнему требуют для своего описания двумерных данных. Естественно предположить, что первые исследователи изображений старались и в этой области найти такие характеристики, которые описывали бы двумерную геометрию изображения при помощи некоторых одномерных массивов. И такие полезные способы описания действительно были найдены. Это так называемые «профили» и «проекции» изображения. Профилем изображения вдоль некоторой линии называется функция интенсивности изображения, распределенного вдоль данной линии (прорезки). Анализ по цвету был реализован при помощи инструмента Line Profile


U8 U16 RGB U32 I16 RGB U64 SGL HSL U32





 **Image** источник изображения

 **Line Coordinates** матрица координат от начала изображения и до его границы

 **Line Graph** функция, определяющая значение пикселя в заданной точке

 **status** имеет значение ИСТИНА (X) , если произошла ошибка или ЛОЖЬ (галочка) , чтобы указать, предупреждение или , что ошибки не было .

 **code** Если состояние имеет значение ИСТИНА , код ненулевой код ошибки . Если состояние имеет значение ЛОЖЬ, код 0 или код предупреждения .

 **source** Источник описывает происхождение ошибки или предупреждения и , в большинстве случаев , название VI или функции , которая привела к ошибке или предупреждение . По умолчанию используется пустая строка .

Глава 3. Экспериментальная часть

3.1. Выбор оптической камеры

Способ передачи данных в компьютер

- Аналоговые
- Веб-камеры(USB-камеры)
- Firewire-камеры(камеры IEEE-1394)
- Сетевые (IP-камеры)

- "Умные" камеры (Smart cameras)

Аналоговые

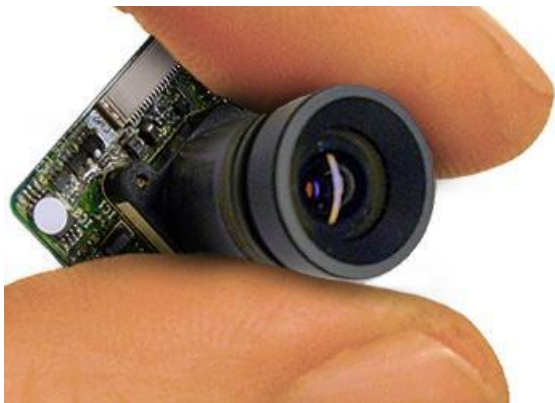


Рисунок 3.1.1 Аналоговая камера LinkSpite

Исторически появились первыми,

сигнал передается в аналоговом сигнале (формат ТВ).

(+) передают данные на большие расстояния, хотя и с помехами (100 м)

(+) легко монтируются, малые размеры

(-) для ввода сигнала в компьютер требуется специальная плата или "ТВ-тюнер", они обычно потребляют много вычислительных ресурсов.

(-) "интерлейс", или черезстрочная развертка - очень затрудняет анализ изображения, если есть движение.

(фактически идет 2 полукадра, каждый 50 раз/сек)

Веб-камеры (USB-камеры)



Рисунок 3.1.2 Вебкамера

Появились в ~2000г.,

передают данные через USB-протокол,

в несжатом виде, либо сжатом в JPEG.

(+) легко подключаются к компьютеру и программному обеспечению

(+) дешевые, имеются в продаже

(-) Накладные расходы - для декодирования JPEG требуются вычислительные ресурсы.

(-) В дешевых моделях обычно плохая оптика и матрица (дает шум на изображении)

(-) Из-за ограничений пропускной способности USB нельзя подключить более 2-х камер к одному USB-хабу, но обычно на PC 2-3 USB хаба.

Firewire-камеры (IEEE-1394)



Рисунок 3.1.3. Скоростная камера

Камеры, передающие сигнал по протоколу FireWire, обычно в пылевлагозащитном корпусе, обычно это камеры для промышленного применения.

(+) передают несжатое видео в отличном качестве на большой скорости

(+) можно подключать несколько камер

(+) обычно имеют отличную оптику

(-) высокая цена

(-) требуют питания, что иногда осложняет подключение к портативным компьютерам

Сетевые (IP-камеры)



Рисунок 3.1.4 IP камера

Камеры, передающие данные по сетевому (проводному или беспроводному) каналу. Сейчас стремительно набирают популярность во всех сферах.

(+) простое подключение к компьютеру

(+) удобство монтажа

(+) возможность передачи данных на неограниченное расстояние, что позволяет конструировать сеть камер, охватывающих здание или район, крепить на дирижабль и т.п.

(+) возможность управления -вращать камеру, настраивать увеличение

(-) могут быть проблемы со скоростью отклика

(-) пока относительно высокая цена

(-) пока недостаточно портативны (2011 год)

"Умные" камеры (Smart cameras)



Рисунок 3.1.5 Смарткамера

Камеры, в корпусе которых располагается компьютер.

Такие камеры являются полнофункциональными системами технического зрения, передающие выходные данные об обнаруженных объектах и т.п. по различным протоколам.

(+) компактность.

(+) масштабируемость -легко строить сети из таких камер.

(-) часто для них требуется адаптация существующих проектов.

(-) дешевые модели достаточно медленные, поэтому хорошо справляются лишь с относительно простыми задачами анализа изображений.

Отдельный тип: Инфракрасные камеры



Рисунок 3.1.6 Инфракрасная камера

Конструируется из обычной камеры путем добавления ИК-фильтра и, зачастую, ИК-подсветки.

+ ИК-лучи почти не видны человеку (в темноте видно как слабый красный цвет), поэтому часто используют для упрощения анализа объектов в поле зрения.

-специализированные ИК-камеры, подходящие для технического зрения, не являются массовым товаром, поэтому их обычно нужно заказывать.

Примеры хороших камер

Sony PS3 Eye



Рисунок 3.1.7 Sony PS3 EYE

320 x 240 : 150 FPS

640 x 480 : 60 FPS

Типы данных:

видимый свет,

ИК (требуется удаление ИК-фильтра)

Цена: 50\$.

USB, CCD

Point Grey Flea3



Рисунок 3.1.8 Point Grey Flea3

648 x 488 : 120 FPS

Тип данных:

-видимый свет,

-ИК (?)

Цена: 600\$.

Модель FL3-FW-03S1C-C

IEEE 1394b, CCD

Microsoft Kinect



Рисунок 3.1.9 Kinect xbox360

640 x 480 : 30 FPS

Тип данных:

видимый свет + глубина

Цена: 150\$.

(глубина -стереозрение с помощью лазерной ИК-подсветки,
поэтому не работает при солнечном свете)

USB, CMOS

Point Grey BumbleBee2



Рисунок 3.1.10Point Grey BumbleBee2

640 x 480 : 48 FPS

Тип данных:

видимый свет + глубина

Цена: 2000\$.

(Глубина -стереозрение с двух камер)

IEEE 1394b, CCD

На основе исследования реально существующих камер был сделан вывод, что наиболее целесообразно для видеорегистрации микробиологических объектов в ридере, использовать мобильный телефон.

3.2 Расчет оптического блока

Расчет оптический блока был проделан для нахождения оптимальных параметров его оптических элементов.

Назначение оптического блока состоит в том, чтобы получить достаточно качественное изображение микроколоний бактерий для дальнейшего его анализа с целью идентификации типа бактерий, образовавших микроколонию. Требование к изображению в разрабатываемой системе не такие высокие, как в оптическом микроскопе, в котором можно увидеть детали строения отдельных бактерий. В оптическом блоке необходимо получить изображение, которое четко передает форму микроколонии, структуру ее краев и ее внутреннюю неоднородность. Под последним подразумевается то, что изображения отдельных микробных тел не будут сливаться в единый фон, а будут различимы и будут давать возможность оценивать их размер.

Приведем характерные размеры изучаемых бактерий:

S.aureus - 1 мкм.

E.coli – 1мкм x 6мкм

C. albicans – 5мкм

При эффективном размере пикселя около 4мкм, необходимо добиться линейного увеличения около 12 раз, для того чтобы объект размером 1мкм содержал порядка 10 (~3*3) пикселей на фотосенсоре.

Оптическая схема для получения изображения в блоке видеорегистрации состоит из двух линз. Для простоты расчета будем считать линзы тонкими.

Расчет оптической такой системы сводится к двукратному применению формулы линзы. Общее линейное увеличение Γ системы из двух линз равно произведению линейных увеличений обеих линз: $\Gamma = \Gamma_1 \cdot \Gamma_2$.

Будем считать также, что объект находится в передней фокальной плоскости первой линзы, а изображение в задней фокальной плоскости второй линзы.

В этом случае можно показать, что общее линейное увеличение системы будет находиться по следующей формуле

$$\Gamma = \frac{F_2}{F_1} \quad (3.2.1)$$

Здесь F_1 и F_2 - фокусные расстояния первой и второй линзы соответственно.

Величина F_2 задается параметрами камеры. При ее ручной установке для съемки объектов на бесконечности $F_2 = 10$ мм. Тогда, из требования $\Gamma = 12$ находим, что $F_1 = 0,83$ мм.

В качестве первой линзы оптического блока была взята линза, входящая в состав стандартного CD привода с фокусным расстоянием $F_1 = 0,8$ мм, что соответствовало требованиям, предъявляемым к оптическому блоку.

3.3 Выбор программной среды (оболочки) для программирования на мобильных устройствах.

Ключевым элементом для реализации компьютерного зрения в режиме реального времени является то, как быстро библиотека компьютерного зрения работает. Здравый смысл подсказывает, что OpenCV должны сокрушить VoofCV, но их разница в том, что OpenCV написано в C / C++ и была доработана с 1999 года, в то время как VoofCV написано в Java и приступила к развитию летом 2011 года. Правда, C / C++ работает быстрее, чем Java, к тому же данный язык является более распространённым. В нашей работе нам потребовалось найти платформу для программирования, поэтому были вынуждены протестировать обе платформы на функциях нужных нам и найти более подходящую.

Исследованные функции:

1. Размытие Гаусса с ядром 5x5
2. Определение контуров методом Собеля с ядром 3x3
3. Определение вершин контуров методом Харисса без сжатия с ядром 5x5
4. Определение контуров методом Канны
5. Нахождение линий методом Хоуга
6. Определение методом SURF

Сравнивая данные мы видим, что алгоритмы низкого уровня был гораздо проще, чем алгоритмы высокого уровня. Обычно низкий уровень не требует каких-либо параметров и имеют более стандартизированные реализации. Алгоритмы высокого уровня открыты для дополнительной интерпретации и часто бывает, что одна библиотека будет иметь параметры конфигурации, не доступные в другой. Для алгоритмов высокого уровня параметры были настроены таким образом, что они произвели подобный вывод. Следует отметить, только скорость рассмотреть, а не качество, за исключением исполнения SURF. Например, одна библиотека может быть намного быстрее, чем другой, но способен достичь такой скорости, производя выход низкого качества.

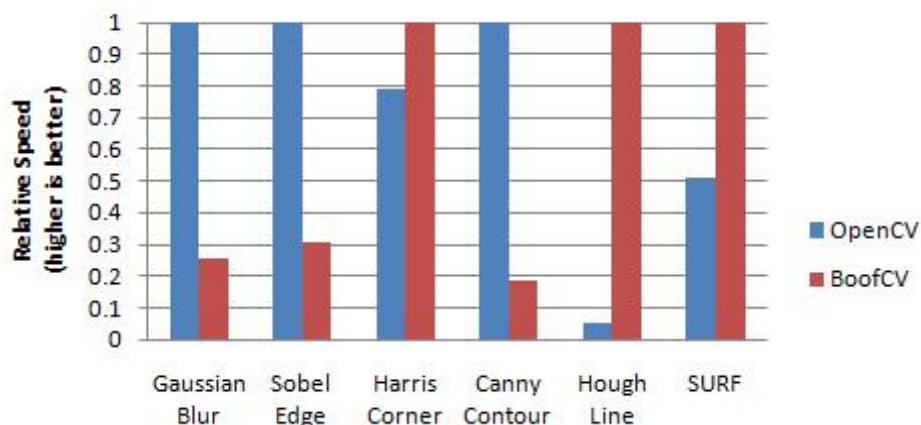


Рисунок 3.3.1 Сравнение скорости работы

Как и ожидалось, OpenCV выполнена очень хорошо для низкого уровня подпрограмм обработки изображений. Эти процедуры могут быть оптимизированы и воспользоваться машины конкретной операции векторизации. Java имеет значительную скорость при прохождении через массивы, потому что

для каждой операции он должен выполнить проверку границ, который не требуется в C / C + +. Также при объявлении новых массивов каждый элемент всегда инициализируется равным нулю, даже если это не требуется. Прошлый опыт показал, что Java имеет тенденцию быть приблизительно в 3 раза медленнее массива тяжелой арифметики, которые эти результаты подтверждают. Для алгоритмов высокого уровня язык, на котором он был реализован, кажется, имеет меньше. Что, скорее всего, самый большой определяющим фактором является алгоритм, реализованный и насколько хорошо задача была выполнена. Это различие реализация может объяснить более высокую производительность BoofCV во многих из операций, несмотря на его незнание языка.

Для всех операций, так как многие итераций были выполнены по крайней мере одна-две и количество операций в секунду (опер / сек) вычисляется. Чтобы избежать низких результатов для очень быстрых операций, количество операций, необходимые для запуска в течение не менее одной секунды автоматически вычисляется. Для целей отображения операций / сек метрика была преобразована в относительно метрики производительности, разделив его на самые эффективные библиотеки.

3.4 Адаптация мобильного устройства для его использования в качестве микроскопической камеры

Функционально устройство микроскопа делится на 3 части:

3.4.1. Осветительная часть

Предназначена для создания светового потока, который позволяет осветить объект таким образом, чтобы последующие части микроскопа предельно точно выполняли свои функции. Осветительная часть микроскопа проходящего света расположена за объектом под объективом в прямых микроскопах (например, биологические, поляризационные и др.) и перед объектом над объективом винвертированных. Осветительная часть конструкции микроскопа включает источник света (лампа и электрический блок питания) и оптико-механическую систему (коллектор, конденсор, полевая и апертурная регулируемые/ириновые диафрагмы).

3.4.2. Воспроизводящая часть

Предназначена для воспроизведения объекта в плоскости изображения с требуемым для исследования качеством изображения и увеличения (т. е. для построения такого изображения, которое как можно точнее и во всех деталях воспроизводило бы объект с соответствующим оптике микроскопа разрешением, увеличением, контрастом и цветопередачей). Воспроизводящая часть обеспечивает первую ступень увеличения и расположена после объекта до плоскости изображения микроскопа. Воспроизводящая часть включает объектив и промежуточную оптическую систему.

Современные микроскопы последнего поколения базируются на оптических системах объективов, скорректированных на бесконечность. Это требует дополнительно применения так называемых тубусных систем, которые параллельные пучки света, выходящие из объектива, «собирают» в плоскости изображения микроскопа.

3.4.3. Визуализирующая часть

Предназначена для получения реального изображения объекта на сетчатке глаза, фотоплёнке или пластинке, на экране телевизионного или компьютерного монитора с дополнительным увеличением (вторая ступень увеличения). Визуализирующая часть расположена между плоскостью изображения объектива и глазами наблюдателя (цифровой камерой). Визуализирующая часть включает монокулярную, биноклярную или тринокулярную визуальную насадку с наблюдательной системой (окулярами, которые работают как лупа). Кроме того, к этой части относятся системы дополнительного увеличения (системы оптовара/смены увеличения); проекционные насадки, в том числе дискуссионные для двух и более наблюдателей; рисовальные аппараты; системы анализа и документирования изображения с соответствующими адаптерами для цифровых камер.

С конструктивно-технологической точки зрения, микроскоп состоит из следующих частей: · механической; · оптической; · электрической.

1. Механическая часть микроскопа

Устройство микроскопа включает себя штатив, который является основным конструктивно-механическим блоком микроскопа. Штатив включает в себя следующие основные блоки: основание и тубусодержатель. Основание представляет собой блок, на котором крепится весь микроскоп и является одной из основных частей микроскопа. В простых микроскопах на основание устанавливают осветительные зеркала или накладные осветители. В более сложных моделях осветительная система встроена в основание без или с блоком питания. Разновидности оснований микроскопа: 1. основание с осветительным зеркалом; 2. так называемое «критическое» или упрощенное освещение; 3. освещение по Келеру.

Тубусодержатель представляет собой блок, часть конструкции микроскопа, на котором закрепляются: 1. фокусировочный механизм грубой и точной настройки микроскопа на резкость — механизм фокусировочного перемещения объективов или столиков; 2. узел крепления сменных предметных столиков; 3. узел крепления сменных насадок (визуальных, фотографических, телевизионных, различных передающих устройств). В микроскопах могут использоваться стойки для крепления узлов (например, фокусировочный механизм в стереомикроскопах или крепление осветителя в некоторых моделях инвертированных микроскопов). Чисто механическим узлом микроскопа является предметный столик, предназначенный для крепления или фиксации в определенном положении объекта наблюдения. Столики бывают неподвижные, координатные и вращающиеся (центрируемые и нецентрируемые).

2. Оптика микроскопа (оптическая часть) Оптические узлы и принадлежности обеспечивают основную функцию микроскопа — создание увеличенного изображения объекта с достаточной степенью достоверности по форме, соотношению размеров составляющих элементов и цвету. Кроме этого, оптика должна обеспечивать такое качество изображения, которое отвечает целям исследования и требованиям методик проводимого анализа. Основными оптическими элементами микроскопа являются оптические элементы, образующие осветительную, наблюдательную (окуляры) и воспроизводящую (в

том числе объективы) системы микроскопа. Объективы микроскопа — представляют собой оптические системы, предназначенные для построения микроскопического изображения в плоскости изображения с соответствующим увеличением, разрешением элементов, точностью воспроизведения по форме и цвету объекта исследования. Объективы являются одними из основных частей микроскопа. Они имеют сложную оптико-механическую конструкцию, которая включает несколько одиночных линз и компонентов, склеенных из 2-х или 3-х линз. Количество линз обусловлено кругом решаемых объективом задач. Чем выше качество изображения, даваемое объективом, тем сложнее его оптическая схема. Общее число линз в сложном объективе может достигать до 14 (например, это может относиться к планапохроматическому объективу с увеличением 100х и числовой апертурой 1,40). Объектив состоит из фронтальной и последующей частей. Фронтальная линза (или система линз) обращена к препарату и является основной при построении изображения соответствующего качества, определяет рабочее расстояние и числовую апертуру объектива. Последующая часть в сочетании с фронтальной обеспечивает требуемое увеличение, фокусное расстояние и качество изображения, а также определяет высоту объектива и длину тубуса микроскопа.

В планобъективах исправлена кривизна изображения по полю, что обеспечивает резкое изображение объекта по всему полю наблюдения. Планобъективы обычно применяются при фотографировании, причем наиболее эффективно применение планапохроматов. Потребность в подобного типа объективах возрастает, однако они достаточно дороги из-за оптической схемы, реализующей плоское поле изображения, и применяемых оптических сред. Поэтому рутинные и рабочие микроскопы комплектуются так называемыми экономичными объективами. К ним относятся объективы с улучшенным качеством изображения по полю: ахростигматы (LEICA), СР-ахроматы и ахропланы (CARL ZEISS), стигмахроматы (ЛОМО). По параметрическим признакам объективы делятся следующим образом: 1. объективы с конечной длиной тубуса (например, 160 мм) и объективы, скорректированные на длину тубуса «бесконечность» (например, с

дополнительной тубусной системой, имеющей фокусное расстояние 160 мм); 2. объективы малых (до 10х); средних (до 50х) и больших (более 50х) увеличений, а также объективы со сверхбольшим увеличением (свыше 100х); 3. объективы малых (до 0,25), средних (до 0,65) и больших (более 0,65) числовых апертур, а также объективы с увеличенными (по сравнению с обычными) числовыми апертурами (например, объективы апохроматической коррекции, а также специальные объективы для люминесцентных микроскопов); 4. объективы с увеличенными (по сравнению с обычными) рабочими расстояниями, а также с большими и сверхбольшими рабочими расстояниями (объективы для работы в инвертированных микроскопах). Рабочее расстояние — это свободное расстояние между объектом (плоскостью покровного стекла) и нижним краем оправы (линзы, если она выступает) фронтального компонента объектива; 5. объективы, обеспечивающие наблюдение в пределах нормального линейного поля (до 18 мм); широкопольные объективы (до 22,5 мм); сверхширокопольные объективы (более 22,5 мм); 6. объективы стандартные (45 мм, 33 мм) и нестандартные по высоте. Высота — расстояние от опорной плоскости объектива (плоскости соприкосновения ввинченного объектива с револьверным устройством) до плоскости предмета при сфокусированном микроскопе, является постоянной величиной и обеспечивает парфокальность комплекта аналогичных по высоте объективов разного увеличения, установленных в револьверном устройстве. Иными словами, если с помощью объектива одного увеличения получить резкое изображение объекта, то при переходе к последующим увеличениям изображение объекта остается резким в пределах глубины резкости объектива. По конструктивно-технологическим признакам существует следующее разделение: 1. объективы, имеющие пружинящую оправу (начиная с числовой апертуры 0,50), и без нее; 2. объективы, имеющие ирисовую диафрагму внутри для изменения числовой апертуры (например, в объективах с увеличенной числовой апертурой, в объективах проходящего света для реализации метода темного поля, в поляризационных объективах отраженного света); 3. объективы с корректирующей (управляющей) оправой, которая

обеспечивает движение оптических элементов внутри объектива (например, для корректировки качества изображения объектива при работе с различной толщиной покровного стекла или с различными иммерсионными жидкостями; а также для изменения увеличения при плавной — панкратической — смене увеличения) и без нее.

По обеспечению методов исследования и контрастирования объективы можно разделить следующим образом: 1. объективы, работающие с покровным и без покровного стекла;

2. объективы проходящего и отраженного света (безрефлексные); люминесцентные объективы (с минимумом собственной люминесценции); поляризационные объективы (без натяжения стекла в оптических элементах, т. е. не вносящие собственную деполяризацию); фазовые объективы (имеющие фазовый элемент — полупрозрачное кольцо внутри объектива); объективы ДИК (DIC), работающие по методу дифференциально-интерференционного контраста (поляризационные с призмным элементом); эпиобъективы (объективы отраженного света, предназначенные для обеспечения методов светлого и темного поля, имеют в конструкции специально рассчитанные осветительные эпи-зеркала); 3. иммерсионные и безиммерсионные объективы. Иммерсия (от лат. *immersio* — погружение) — жидкость, заполняющая пространство между объектом наблюдения и специальным иммерсионным объективом (конденсором и предметным стеклом). В основном применяются три типа иммерсионных жидкостей: масляная иммерсия (МИ/Oil), водная иммерсия (ВИ/W) и глицериновая иммерсия (ГИ/Glyc), причем последняя в основном применяется в ультрафиолетовой микроскопии. Иммерсия применяется в тех случаях, когда требуется повысить разрешающую способность микроскопа или её применения требует технологический процесс микроскопирования. При этом происходит: 1. повышение видимости за счет увеличения разности показателя преломления среды и объекта; 2. увеличение глубины просматриваемого слоя, который зависит от показателя преломления среды. Кроме того, иммерсионная жидкость может уменьшать количество рассеянного света за счет исчезновения

бликов от объекта. При этом устраняются неизбежные потери света при его попадании в объектив. Иммерсионные объективы. Качество изображения, параметры и оптическая конструкция иммерсионных объективов рассчитываются и выбираются с учетом толщины слоя иммерсии, которая рассматривается как дополнительная линза с соответствующим показателем преломления. Иммерсионная жидкость, расположенная между объектом и фронтальным компонентом объектива, увеличивает угол, под которым рассматривается объект (апертурный угол). Числовая апертура безыммерсионного (сухого) объектива не превышает 1,0 (разрешающая способность порядка 0,3 мкм для основной длины волны); иммерсионного — доходит до 1,40 в зависимости от показателя преломления иммерсии и технологических возможностей изготовления фронтальной линзы (разрешающая способность такого объектива порядка 0,12 мкм). Иммерсионные объективы больших увеличений имеют короткое фокусное расстояние — 1,5–2,5 мм при свободном рабочем расстоянии 0,1–0,3 мм (расстояние от плоскости препарата до оправы фронтальной линзы объектива). Маркировка объективов. Данные о каждом объективе маркируются на его корпусе с указанием следующих параметров: 1. увеличение («х»-крат, раз): 8х, 40х, 90х; 2. числовая апертура: 0,20; 0,65, пример: 40/0,65 или 40х/0,65; 3. дополнительная буквенная маркировка, если объектив используется при различных методах исследования и контрастирования: фазовый — Ф (Рп2 — цифра соответствует маркировке на специальном вкладыше), поляризационный — П (Pol), люминесцентный — Л (L), фазово-люминесцентный — ФЛ (PhL), ЭПИ (Epi, HD) — эпиобъектив для работы в отраженном свете по методу темного поля, дифференциально-интерференционный контраст — ДИК (DIC), пример: 40х/0,65 Ф или Ph2 40х/0,65; 4. маркировка типа оптической коррекции: апохромат — АПО (APO), планахромат — ПЛАН (PL, Plan), планапохромат — ПЛАН-АПО (Plan- Apo), улучшенный ахромат, полуплан — СХ — стигмахромат (Achrostigmat, CP-achromat, Achroplan), микрофлюар (полуплан- полуапохромат) — СФ или М-ФЛЮАР (MICROFLUAR, NEOFLUAR, NPL, FLUOTAR).

Осветительная система

Осветительная система является важной частью конструкции микроскопа и представляет собой систему линз, диафрагм и зеркал (последние применяются при необходимости), обеспечивающую равномерное освещение объекта и полное заполнение апертуры объектива. Осветительная система микроскопа проходящего света состоит из двух частей — коллектора и конденсора. Коллектор. При встроенной осветительной системе проходящего света коллекторная часть расположена вблизи источника света в основании микроскопа и предназначена для увеличения размера светящегося тела. Для обеспечения настройки коллектор может быть выполнен подвижным и перемещаться вдоль оптической оси. Вблизи коллектора располагается полевая диафрагма микроскопа. Конденсор. Оптическая система конденсора предназначена для увеличения количества света, поступающего в микроскоп. Конденсор располагается между объектом (предметным столиком) и осветителем (источником света). Чаще всего в учебных и простых микроскопах конденсор может быть выполнен несъемным и неподвижным. В остальных случаях конденсор является съемной частью и при настройке освещения имеет фокусирующее перемещение вдоль оптической оси и центрирующее перемещение, перпендикулярное оптической оси. При конденсоре всегда находится осветительная апертурная ирисовая диафрагма.

Электрическая часть микроскопа В современных микроскопах, вместо зеркал, используются различные источники освещения, питаемые от электрической сети. Это могут быть как обычные лампы накаливания, так и галогенные, и ксеноновые, и ртутные лампы. Также все большую популярность набирают светодиодные осветители. Они обладают значительными преимуществами перед обычными лампами, как например долговечность, меньшее энергопотребление и др. Для питания источника освещения используются различные блоки питания, блоки розжига и другие устройства, преобразующие ток из электрической сети в подходящий для питания того или иного источника освещения. Также это могут быть и аккумуляторные батареи, что позволяет использовать микроскопы в

полевых условиях при отсутствии точки подключения. Разрешающая способность микроскопа дает отдельное изображение двух близких друг другу линий. Невооруженный человеческий глаз имеет разрешающую способность около $1/10$ мм или 100 мкм. Лучший световой микроскоп примерно в 500 раз улучшает возможность человеческого глаза, т. е. его разрешающая способность составляет около 0,2 мкм или 200 нм. Разрешающая способность и увеличение не одно и то же. Можно получить большое увеличение, но не улучшить его разрешение. Различают полезное и бесполезное увеличения. Под полезным понимают такое увеличение наблюдаемого объекта, при котором можно выявить новые детали его строения. Бесполезное - это увеличение, при котором, увеличивая объект в сотни и более раз, нельзя обнаружить новых деталей строения. В учебных лабораториях обычно используют световые микроскопы, на которых микропрепараты рассматриваются с использованием естественного или искусственного света. Наиболее распространены световые биологические микроскопы: БИОЛАМ, МИКМЕД, МБР, МБИ и МБС. Они дают увеличение в пределах от 56 до 1350 раз. Объектив - определяет полезное увеличение объекта. Объектив состоит из нескольких линз. Увеличение объектива обозначено на нем цифрами. В учебных целях используют обычно объективы $\times 8$ и $\times 40$. Окуляр состоит из 2-3 линз. Увеличение окуляров обозначено на них цифрами: $\times 7$, $\times 10$, $\times 15$. Окуляры не выявляют новых деталей строения и в этом отношении их увеличение бесполезно. Для определения общего увеличения микроскопа следует умножить увеличение объектива на увеличение окуляра. В случае использования бинокулярной или тринокулярной насадки, в данное уравнение нужно добавить собственное увеличение насадки. Для бинокулярной насадки АУ-12 ЛОМО, увеличение которой составляет 1,5X. В таких насадках как АУ-26 или МФН-11 увеличение можно менять, собственное сменное увеличение насадки АУ-26 -- 1,1x; 1,6x и 2,5x Осветительное устройство состоит из зеркала или электроосветителя, конденсора с ирисовой диафрагмой и светофильтром, расположенных под предметным столиком. Они предназначены для освещения объекта пучком света. Механическая система микроскопа состоит из подставки,

коробки с микрометрическим механизмом и микрометрическим винтом, тубусодержателя, винта грубой наводки, кронштейна конденсора, винта перемещения конденсора, револьвера, предметного столика. Правила работы с микроскопом При работе с микроскопом необходимо соблюдать операции в следующем порядке:

1. Работать с микроскопом следует сидя; 2. Микроскоп осмотреть, вытереть от пыли мягкой салфеткой объективы, окуляр, зеркало; 3. Микроскоп установить перед собой, немного слева на 2-3 см от края стола. Во время работы его не сдвигать; 4. Открыть полностью диафрагму, поднять конденсор в крайнее верхнее положение; 5. Работу с микроскопом всегда начинать с малого увеличения; 6. Опустить объектив 8 х в рабочее положение, т. е. на расстояние 1 см от предметного стекла; 7. Глядя одним глазом в окуляр и пользуясь зеркалом с вогнутой стороной, направить свет от окна в объектив, а затем максимально и равномерно осветить поле зрения; 8. Положить микропрепарат на предметный столик так, чтобы изучаемый объект находился под объективом. Глядя сбоку, опускать объектив при помощи макровинта до тех пор, пока расстояние между нижней линзой объектива и микропрепаратом не станет 4-5 мм ; 9. Смотреть одним глазом в окуляр и вращать винт грубой наводки на себя, плавно поднимая объектив до положения, при котором хорошо будет видно изображение объекта. Нельзя смотреть в окуляр и опускать объектив. Фронтальная линза может раздавить покровное стекло, и на ней появятся царапины;
10. Передвигая препарат рукой, найти нужное место, расположить его в центре поля зрения микроскопа; 11. Если изображение не появилось, то надо повторить все операции пунктов 6, 7, 8, 9; 12. Для изучения объекта при большом увеличении сначала нужно поставить выбранный участок в центр поля зрения микроскопа при малом увеличении. Затем поменять объектив на 40 х, поворачивая револьвер, так чтобы он занял рабочее положение. При помощи микрометрического винта добиться хорошего изображения объекта. На коробке микрометрического механизма имеются две риски, а на микрометрическом винте - точка, которая должна все время находиться между рисками. Если она выходит за

их пределы, ее необходимо вернуть в нормальное положение. При несоблюдении этого правила, микрометрический винт может перестать действовать; 13. По окончании работы с большим увеличением, установить малое увеличение, поднять объектив, снять с рабочего столика препарат, протереть чистой салфеткой все части микроскопа, накрыть его полиэтиленовым пакетом и поставить в шкаф.

Фотографии микроскопа



Рисунок 3.4.1 Микроскоп



Рисунок 3.4.2 Микроскоп вид сверху

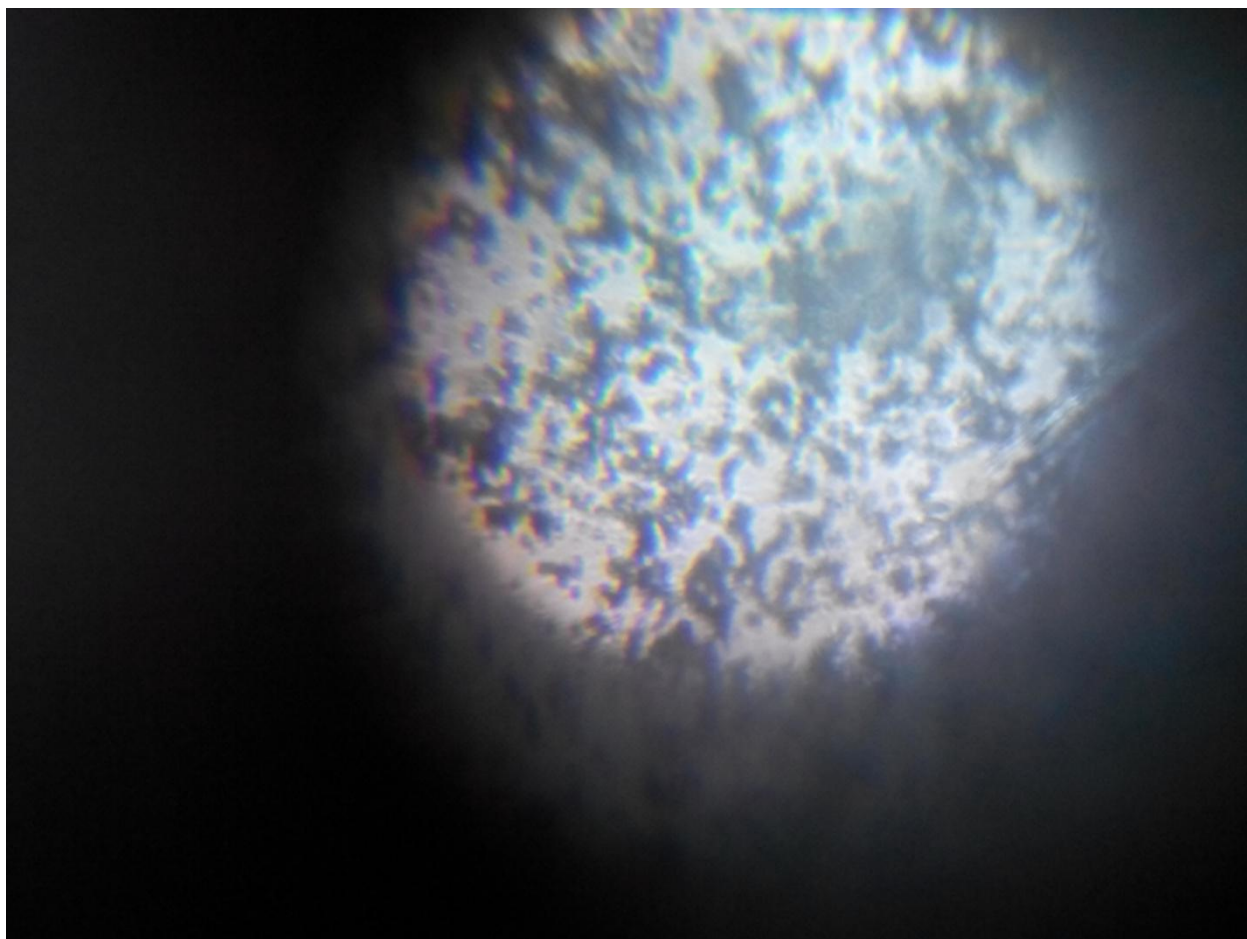


Рисунок 3.4.3 Изображение с модельными объектами

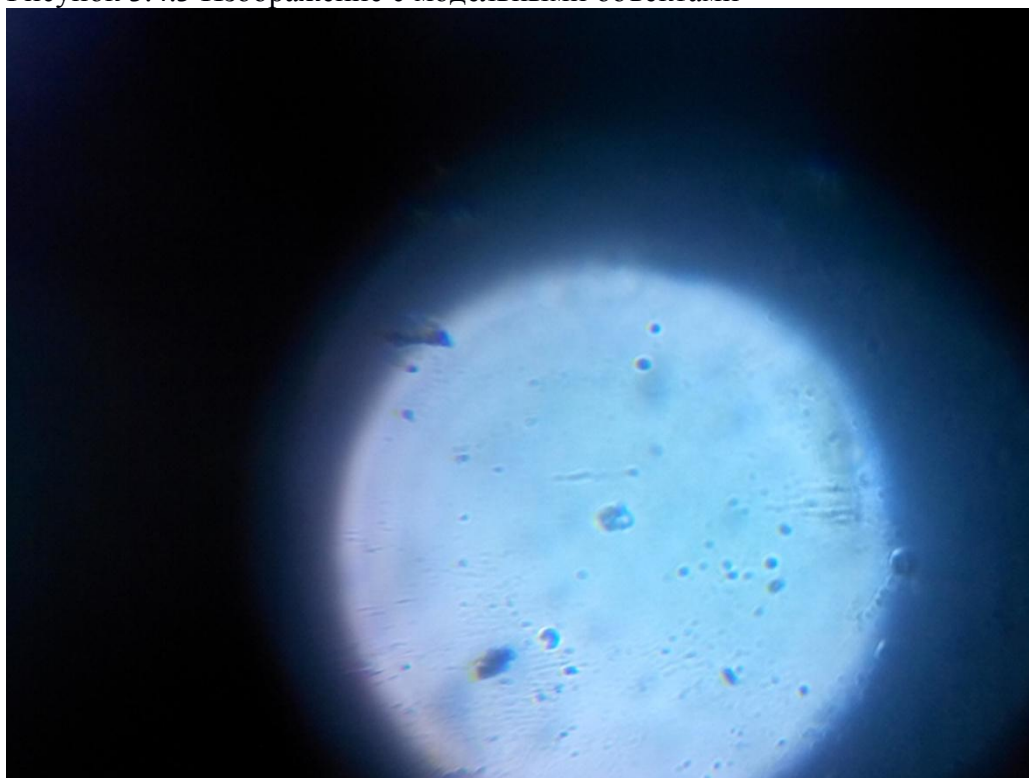


Рисунок 3.4.4 Исследуемый порошок № 2

Глава 4 Результаты и обсуждение

В настоящей дипломной работе произведена разработка модуля распознавания образов в лаборатории на чипе, включающего в себя программное обеспечение для распознавания образов бактерий и блок видеорегистрации с микрокапиллярным чипом для анализируемой пробы, выполненный по правилам миниатюризации

Разработан макет блока видеорегистрации на базе смартфона с фотокамерой 13 МПикс . Блок испытан на модельном объекте и показана возможность его практической реализации в реальном устройстве регистрации в лаборатории на чипе.

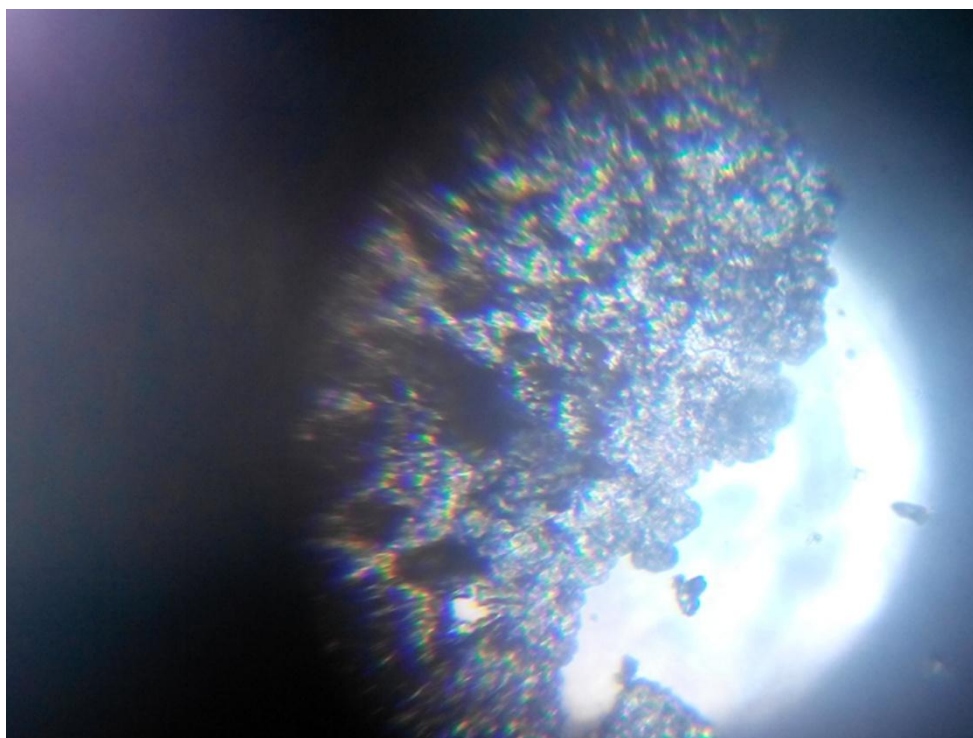


Рисунок 4.1 Плесень на апельсине

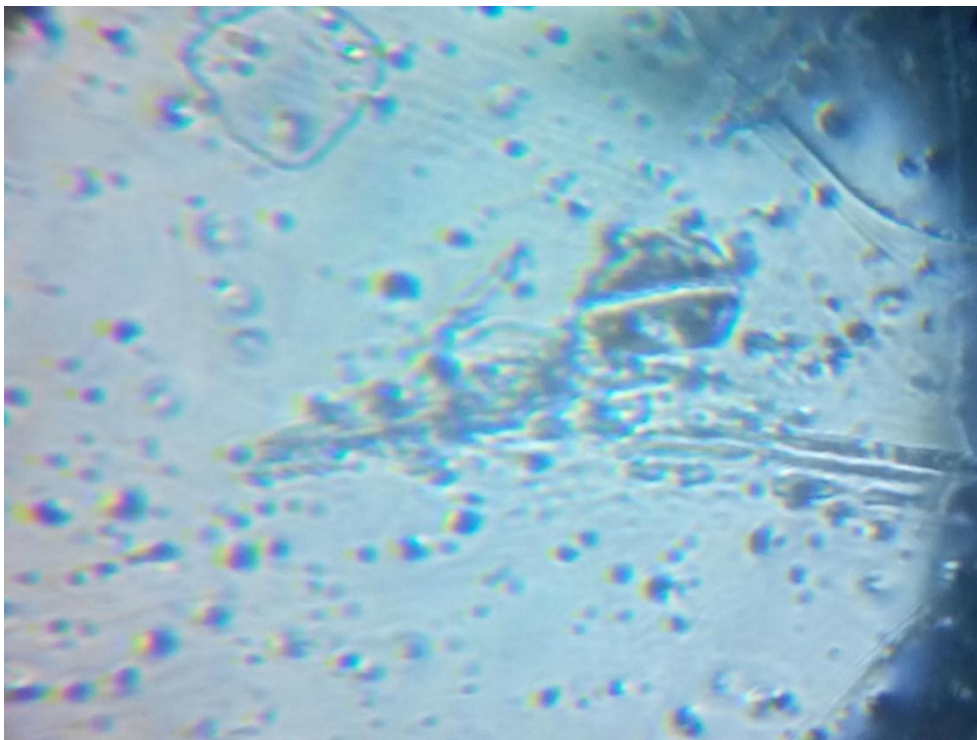


Рисунок 4.2 Порошок

Разработан алгоритм распознавания бактерий на основе концепции анализа пространства признаков и реализован программный комплекс для распознавания, как отдельных бактерий, так и их микроколоний. Результаты работы программы на смартфоне:

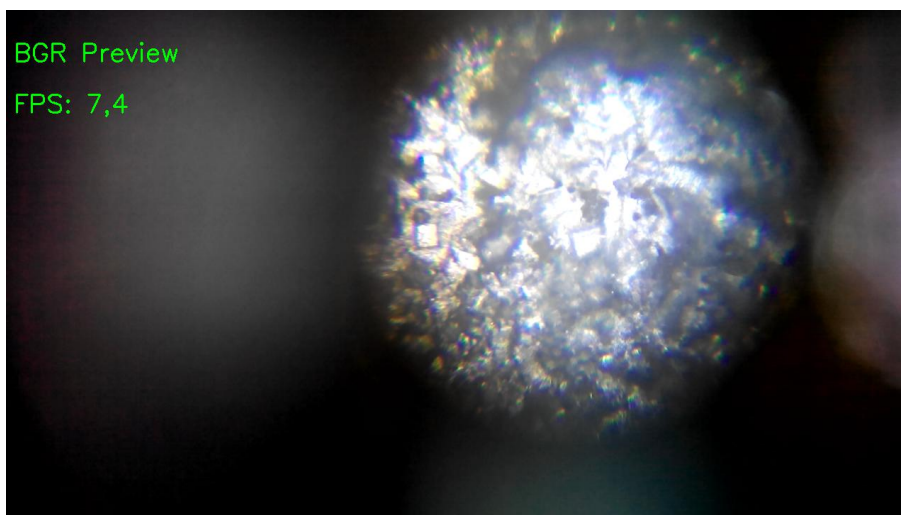


Рисунок 4.3. Модельный объект №1



Рисунок 4.4 Алгоритм Канни для объекта 1

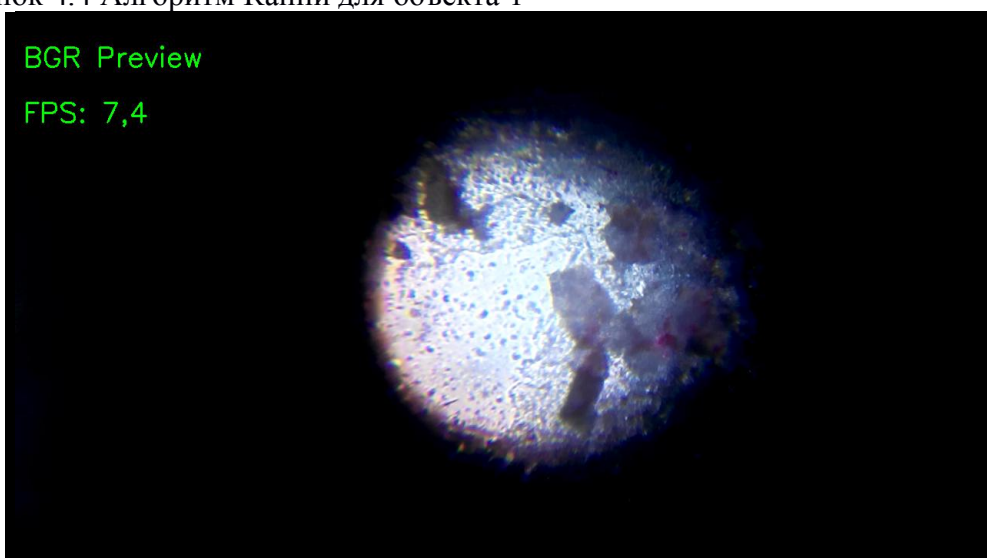


Рисунок 4.5 Модельный объект №2

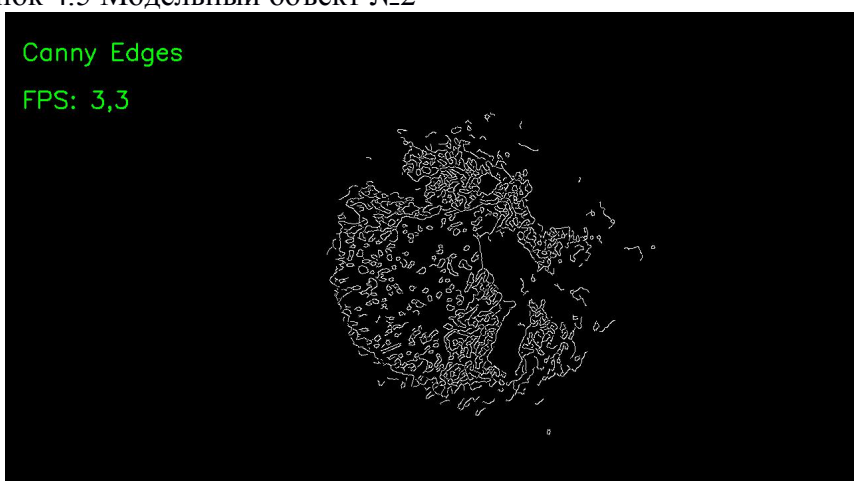


Рисунок 4.6 Реализация алгоритма Канни



Рисунок 4.7 Модельный порошок №3

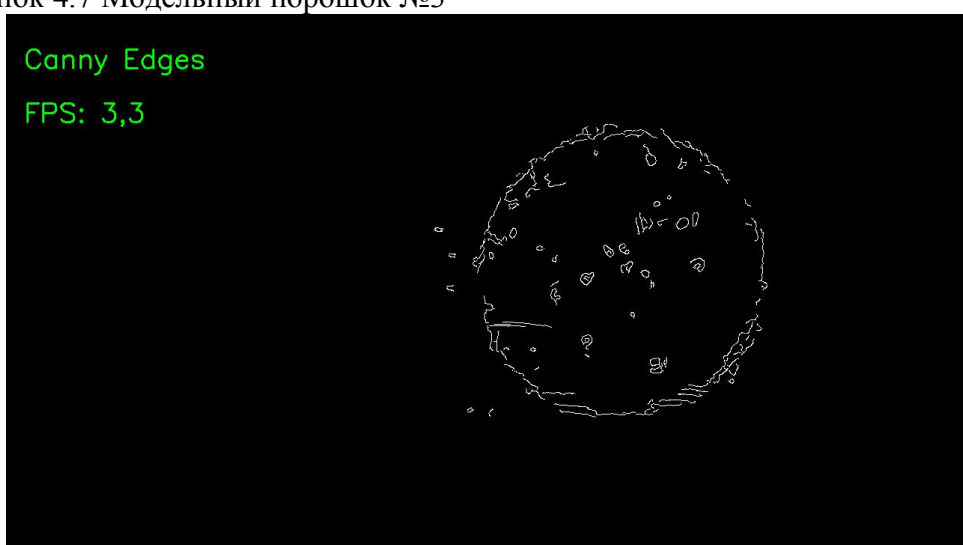


Рисунок 4.8 Алгоритм Канни на объекте №3

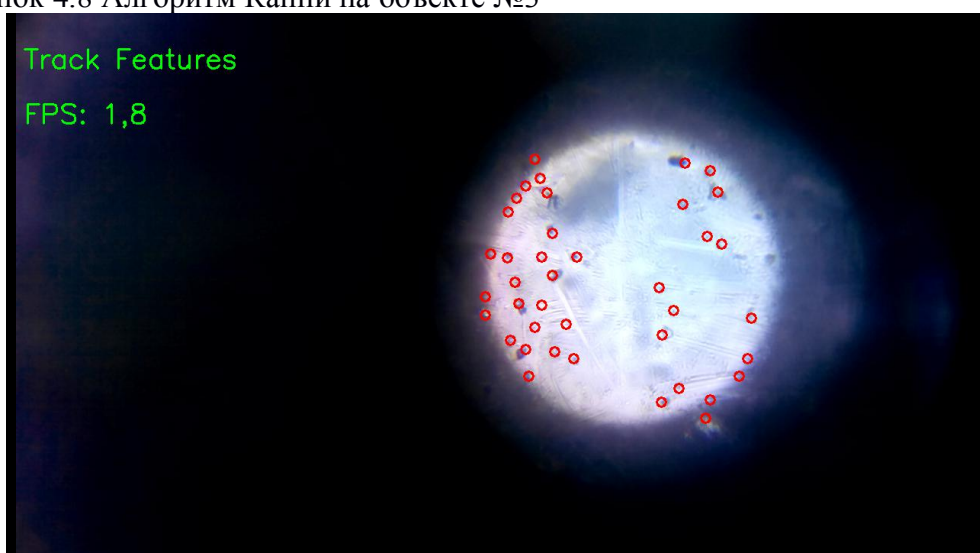


Рисунок 4.9 Нахождение характерных точек

Работа программного комплекса испытана на модельных объектах и на реальных микробиологических объектах, таких как *E.coli*, *Candida albicans* и *S.aureus*. Найдены количественные признаки биообъектов, позволяющие идентифицировать, как различные бактерий, так и их колонии.

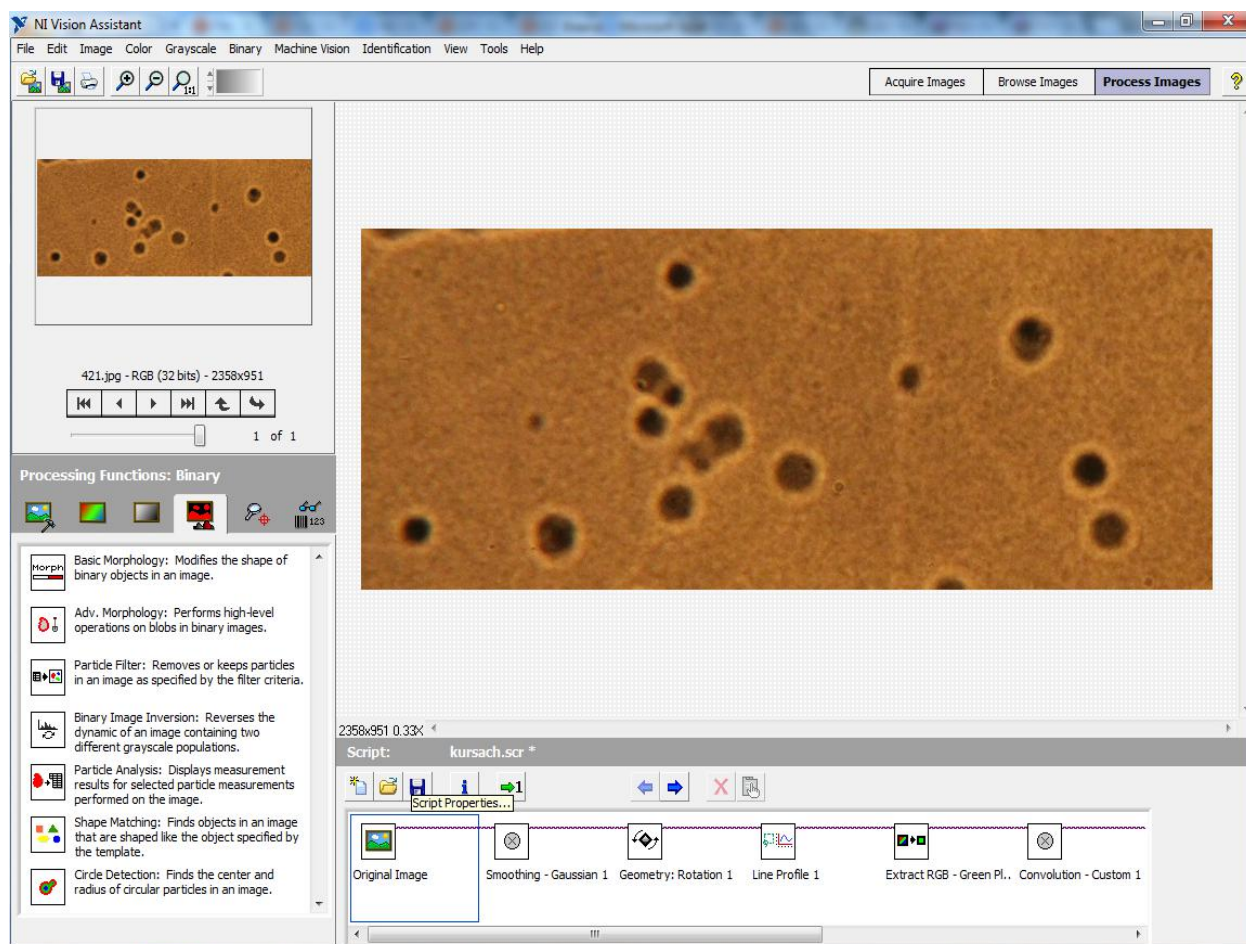
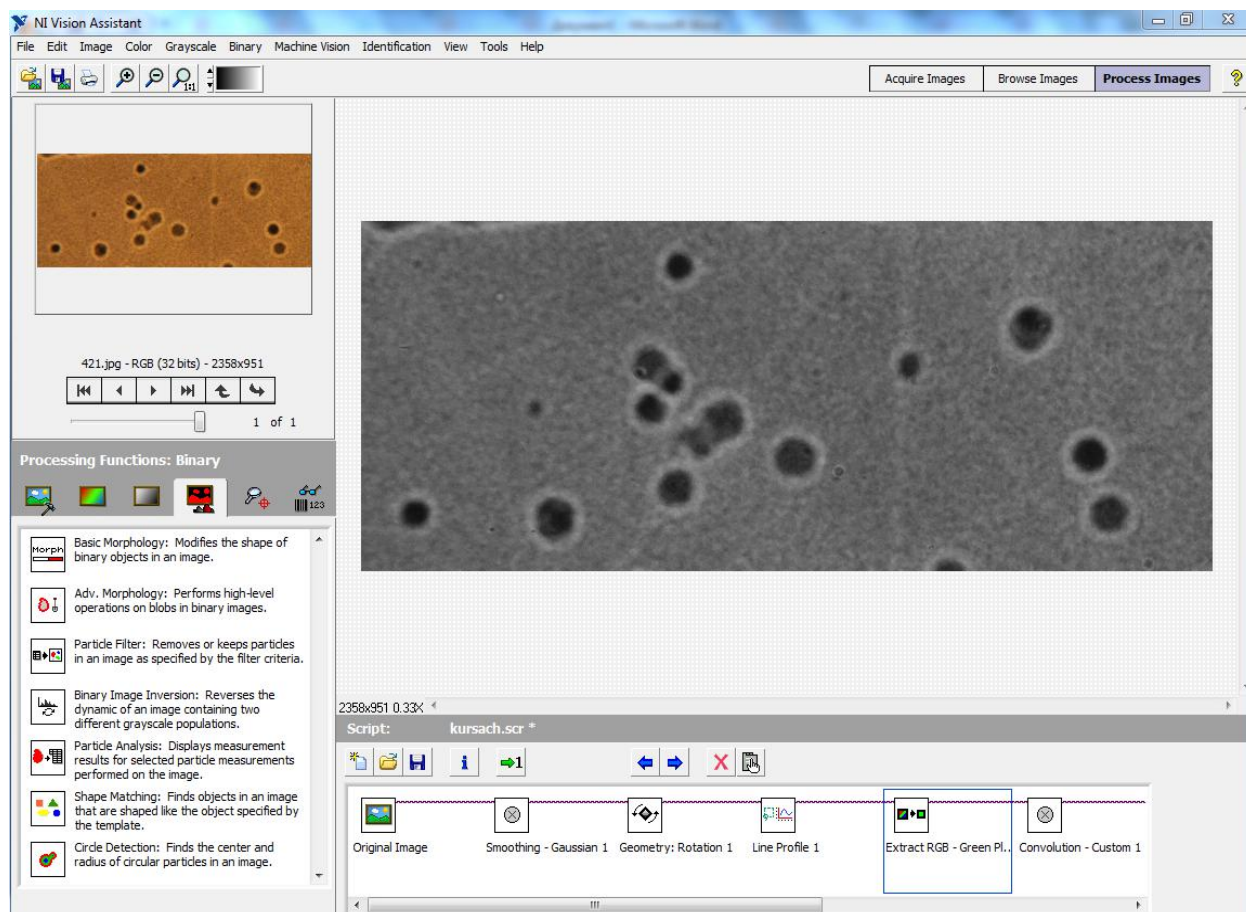


Рисунок 4.10 Работа с изображениями микроколоний *S.aureus* (увеличение X 100 раз)



Р

Рисунок 4.11 Перевод оригинального изображения в оттенки серого по формуле чёрно-белого телевизора

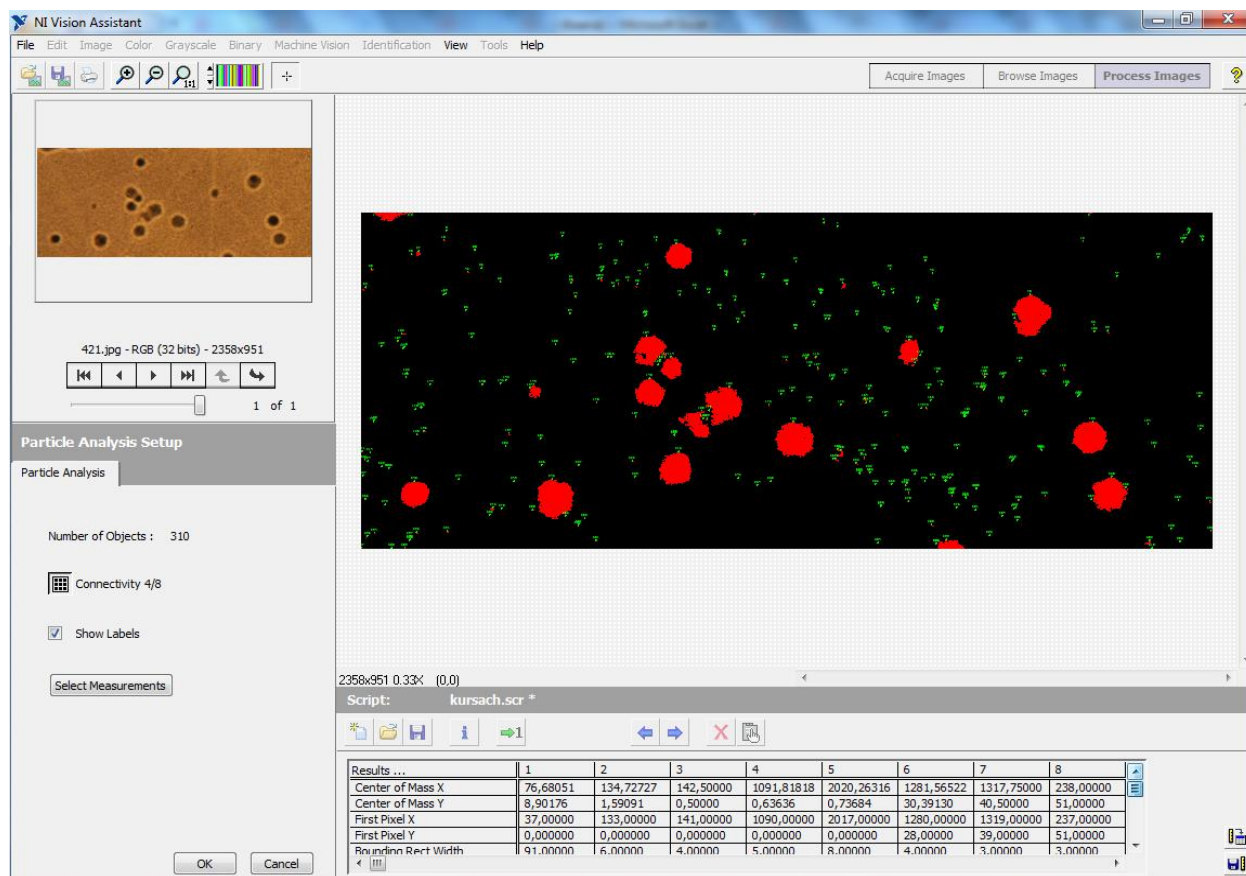


Рисунок 4.12 Маркинг объектов на изображении

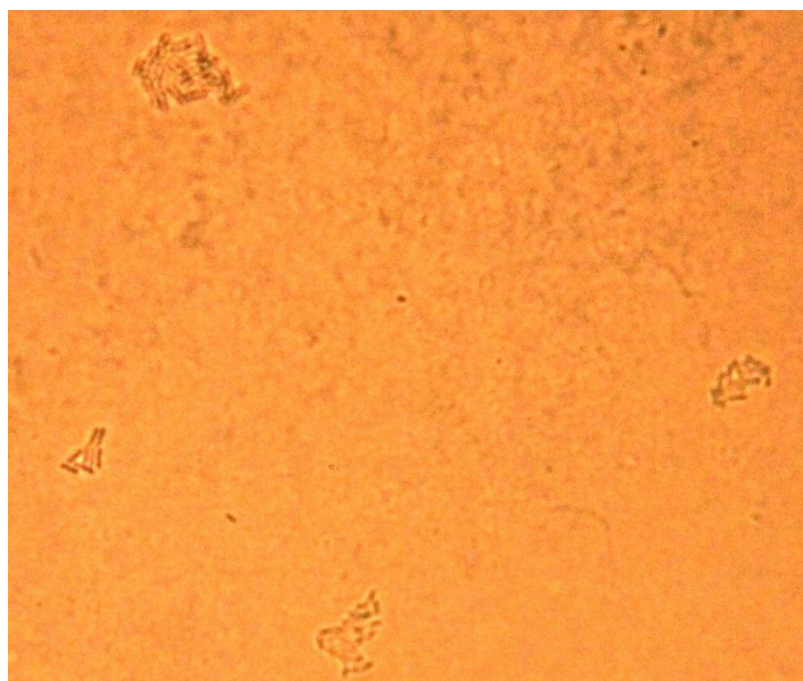


Рисунок 4.13 Микрocolонии E.coli первый образец

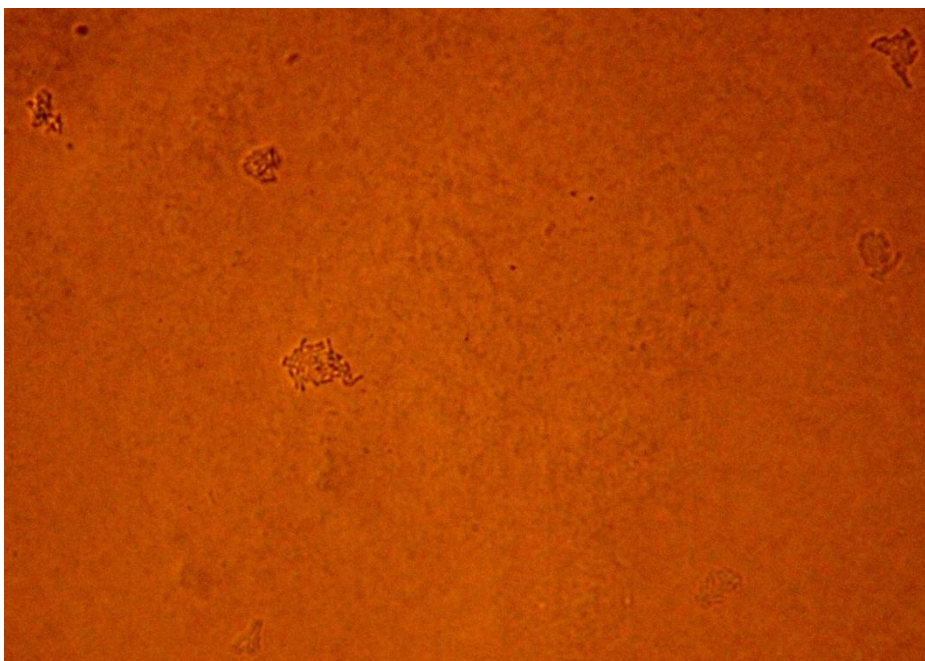


Рисунок 4.14 Микроколони́и *E.coli* второй образец



Рисунок 4.15 Микроколони́и *Candida albicans* первый образец

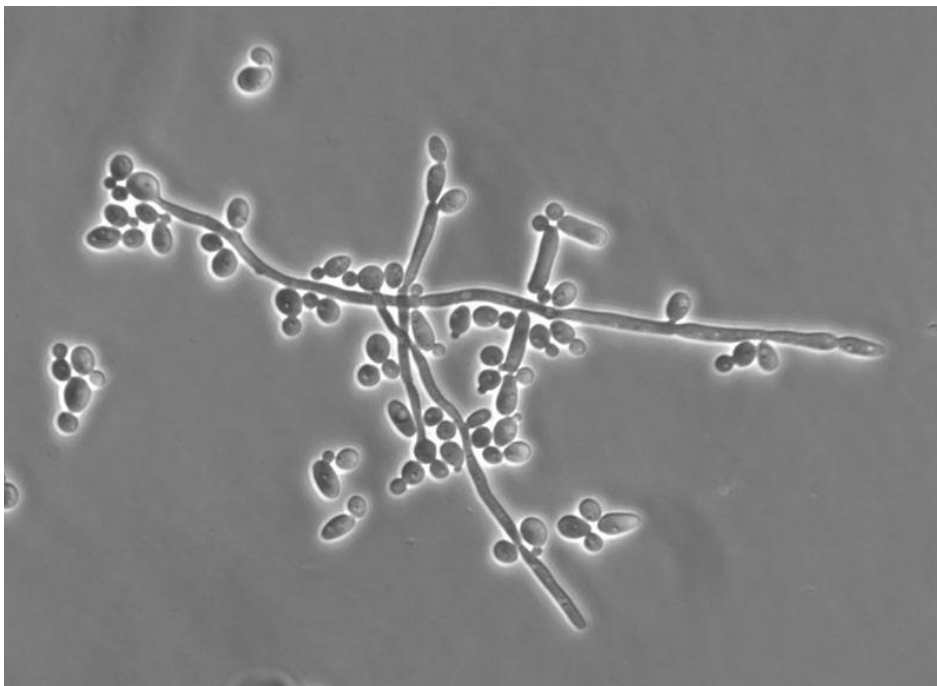


Рисунок 4.16 Микроколонии *Candida albicans* второй образец

Проведя эксперименты мы убедились, что микроколонии различимы.

На основе расчётов были выделены три признака для цветовой идентификации и три для геометрической. Для геометрической: коэффициент некруглости-отношение Фере диаметров, отклонение от единицы- $4\pi \cdot S/P^2$, а также изрезанность контура - отношение реального периметра к аппроксимированному с помощью эллипса. Результат приведён ниже на графиках.

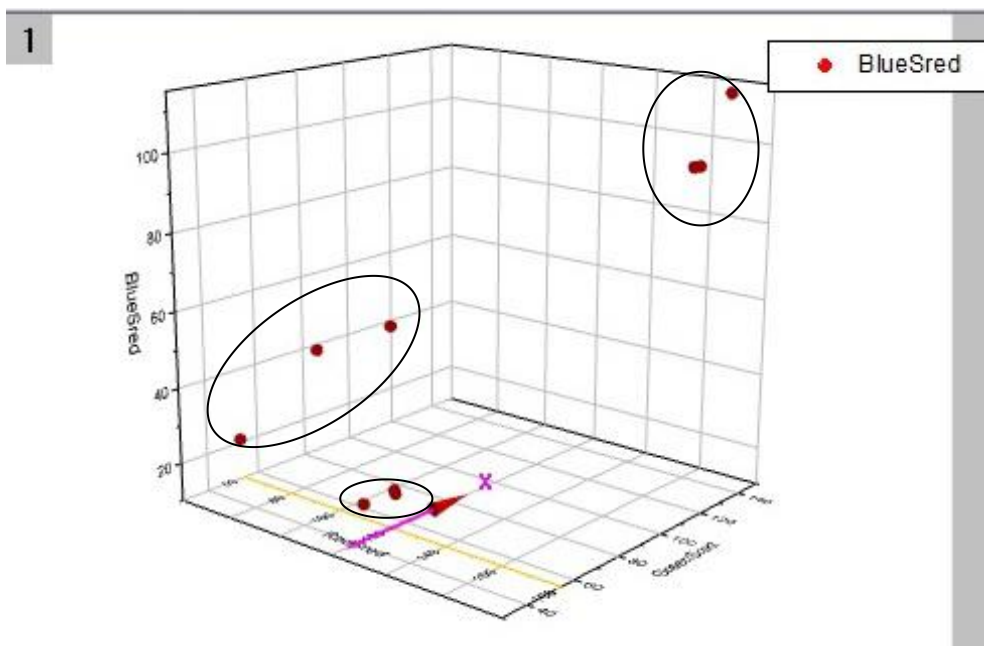


Рисунок 4.17 Распознавание по цвету

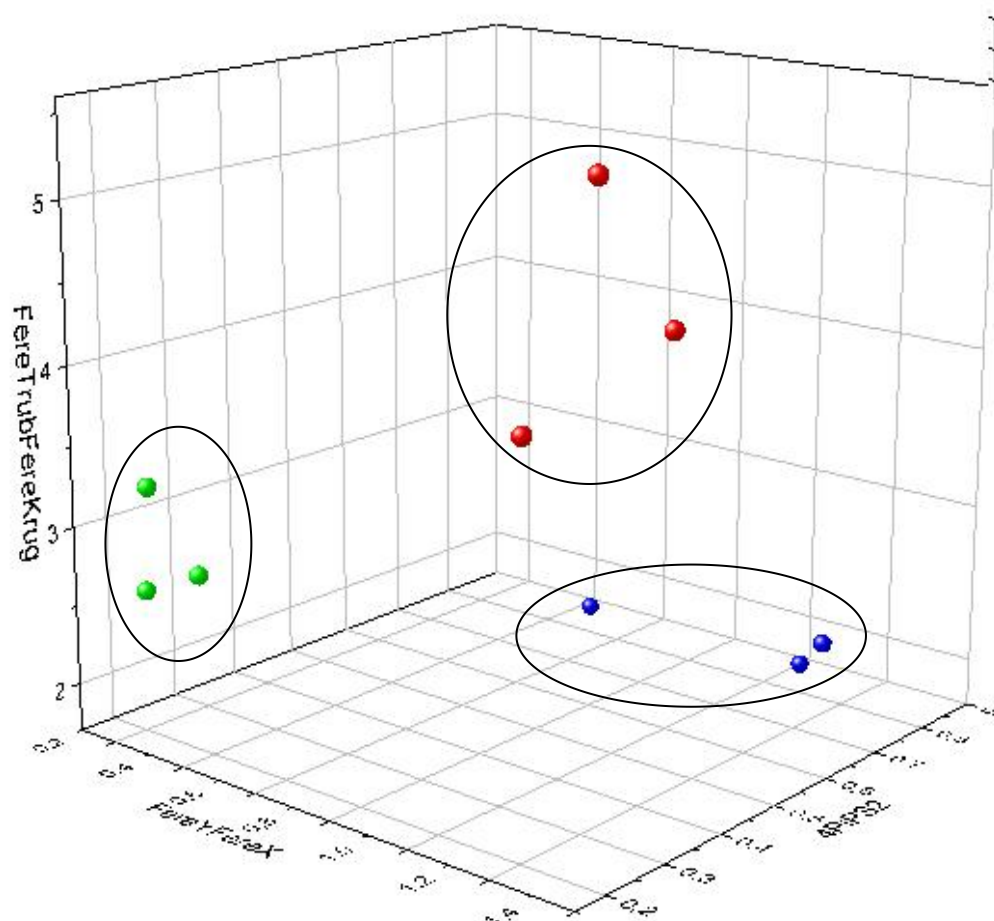


Рисунок 4.18. Распознавание по геометрическим признакам.

На основе данных графиков можно сделать выводы по тому к какому виду относится данная микроколония. Если объект относится к определённом виду микроколонии, то он будет входить в область эллипса, соответствующего данной колонии. Соответствующие микроколонии на рисунок окрашены цветами для идентификации пользователем.

Созданные программные коды могут применяться как элементы программного обеспечения разрабатываемых в настоящее время лабораторий на чипе.

Заключение

В настоящей дипломной работе произведена разработка и миниатюризация модуля распознавания образов в лаборатории на чипе, включающего в себя программное обеспечение для распознавания образов бактерий, а также блок видеорегистрации, включающий микроскоп на основе смартфона для анализируемой пробы.

Разработан макет блока видеорегистрации на базе смартфона Lenovo с фотокамерой 13 Мпикс. Блок испытан на модельном объекте и показана возможность его практической реализации в реальном устройстве регистрации в лаборатории на чипе.

Разработан алгоритм распознавания бактерий на основе концепции анализа пространства признаков и реализован программный комплекс для распознавания, как отдельных бактерий, так и их микроколоний при помощи персонального компьютера и программы LabView с модулем Imaq Vision, а также средства первичной обработки на основе смартфона с установленной библиотекой OpenCV.

Работа программного комплекса испытана на модельных объектах и на реальных микробиологических объектах, таких как *E.coli*, *Staphylococcus aureus* и *Candida albicans*. Найдены количественные признаки биообъектов, позволяющие идентифицировать, как различные бактерий, так и их колонии. Созданные программные коды могут применяться как элементы программного обеспечения разрабатываемых в настоящее время лабораторий на чипе.

Список используемой литературы

1. Ортогональные преобразования при цифровой обработке сигналов. Ахмед Н.
2. Статистические методы распознавания образов Ю. Лифшиц
3. Математические методы распознавания образов Курс лекций МГУ, ВМиК, кафедра «Математические методы прогнозирования» © Местецкий Леонид Моисеевич, 2002–2004
4. Android and OpenCV Tutorial. Computer Vision Lab Tutorial 26 September 2013 Lorenz Meier, Amaël Delaunoy, Kalin Kolev
5. OpenCv & Android Working with imagesBadrinath Kulkarni GDG Bangalor
6. И.С.Грузман, В.С. Киричук, В.П. Косых, Г.И. Перетягин, А.А.Спектор
ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ В ИНФОРМАЦИОННЫХ СИСТЕМАХ
7. Picture processing and digital filtering Huang
8. Апальков И.В., Хрящев В.В. Удаление шума из изображений на основе нелинейных алгоритмов с использованием ранговой статистики. — Ярославский государственный университет, 2007.
9. Бухтояров С. С., Удаление шума из изображений нелинейными цифровыми фильтрами на основе ранговой статистики. – Москва, 2007.
10. Гонсалес Р., Вудс Р. Цифровая обработка изображений. – М.: Техносфера, 2005
11. Грузман И.С, В.С. Киричук, В.П. Косых, Г.И. Перетягин, А.А.Спектор. Цифровая обработка изображений в информационных системах: Учебное пособие.— Новосибирск: Изд-во НГТУ, 2000. — 168.
12. Калинкина Д., Д. Ватолин, Проблема подавления шума на изображениях и видео и различные подходы к ее решению, <http://cgm.computergraphics.ru>

13. Фисенко В.Т., Фисенко Т.Ю., Компьютерная обработка и распознавание изображений: учеб. пособие. — СПб: СПбГУ ИТМО, 2008. — 192 с.
14. Buades, A., B. Coll, and J.M. Morel, "A review of image denoising algorithms, with a new one," SIAM Multiscale Modeling and Simulation, vol. 4, pp. 490-530, 2005.
15. Katkovnik, V., K. Egiazarian, and J. Astola, "Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule", J. of Math. Imaging and Vision, vol. 16, no. 3, pp. 223-235, 2002.
16. Katkovnik, V., K. Egiazarian, and J. Astola, Local Approximation Techniques in Signal and Image Processing, SPIE Press, Monograph Vol. PM157, September 2006.
17. Katkovnik, V., A. Foi, K. Dabov, and K. Egiazarian, "Spatially adaptive support as a leading model-selection tool for image filtering", Proc. First Workshop Inf. Th. Methods Sci. Eng., WITMSE, Tampere, August 2008.
18. Rudin, L., S Osher, E Fatemi, "Nonlinear total variation based noise removal algorithms", Physica D, 1992 — csee.wvu.edu27
19. Welch G., G. Bishop, An Introduction to the Kalman Filter, University of North Carolina at Chapel Hill, Chapel Hill, NC, 1995
20. А.А. Дудкин, Д.А. Вершок, А.М. Селиханович Объединенный институт проблем информатики НАН Беларуси, г. Минск, Беларусь, doudkin@newman.bas-net.by Выделение контуров на полутонных изображениях топологических слоев интегральных схем
21. Панов В. А., Андреев Л. Н. Оптика микроскопов. Расчет и проектирование. Л., «Машиностроение» (Ленингр. отделение), 1976.

Приложение 1

```
/**
*****

/** WARNING: This file was automatically generated. Any changes you make *
/** to this file will be lost if you generate the file again. *
/**
*****

#include <stdlib.h>
#include <string.h>
#include <math.h>
#include <nivision.h>
#include <nimachinevision.h>
#include <windows.h>

// If you call Machine Vision functions in your script, add NIMachineVision.c to the
project.

#define IVA_MAX_BUFFERS 10
#define IVA_STORE_RESULT_NAMES

#define VisionErrChk(Function) {if (!(Function)) {success = 0; goto Error;}}

typedef enum IVA_ResultType_Enum {IVA_NUMERIC, IVA_BOOLEAN,
IVA_STRING} IVA_ResultType;
```

```
typedef union IVA_ResultValue_Struct    // A result in Vision Assistant can be of type
double, BOOL or string.
```

```
{
    double numVal;
    BOOL  boolVal;
    char* strVal;
} IVA_ResultValue;
```

```
typedef struct IVA_Result_Struct
{
#if defined (IVA_STORE_RESULT_NAMES)
    char resultName[256];    // Result name
#endif
    IVA_ResultType type;    // Result type
    IVA_ResultValue resultVal;    // Result value
} IVA_Result;
```

```
typedef struct IVA_StepResultsStruct
{
#if defined (IVA_STORE_RESULT_NAMES)
    char stepName[256];    // Step name
#endif
    int    numResults;    // number of results created by the step
    IVA_Result* results;    // array of results
} IVA_StepResults;
```

```
typedef struct IVA_Data_Struct
{
    Image* buffers[IVA_MAX_BUFFERS];    // Vision Assistant Image Buffers
    IVA_StepResults* stepResults;    // Array of step results
}
```

```

    int numSteps;                // Number of steps allocated in the stepResults
array
    CoordinateSystem *baseCoordinateSystems; // Base Coordinate Systems
    CoordinateSystem *MeasurementSystems;    // Measurement Coordinate Systems
    int numCoordSys;                // Number of coordinate systems
} IVA_Data;

```

```

static IVA_Data* IVA_InitData(int numSteps, int numCoordSys);
static int IVA_DisposeData(IVA_Data* ivaData);
static int IVA_DisposeStepResults(IVA_Data* ivaData, int stepIndex);
static int IVA_LineProfile_Color(Image* image,
                                ROI* roi,
                                int mode);
static int IVA_CLRExtractValue(Image* image);
static int IVA_Particle(Image* image,
                        int connectivity,
                        int pPixelMeasurements[],
                        int numPixelMeasurements,
                        int pCalibratedMeasurements[],
                        int numCalibratedMeasurements,
                        IVA_Data* ivaData,
                        int stepIndex);

```

```

int IVA_ProcessImage(Image *image)
{
    int success = 1;
    IVA_Data *ivaData;
    ROI *roi;

```

```

float kernel[9] = {-1,-1,-1,-1,10,-1,-1,-1,-1};
int pKernel[9] = {1,1,1,1,1,1,1,1,1};
StructuringElement structElem;
int pKernel1[9] = {0,1,0,1,1,1,0,1,0};
StructuringElement structElem1;
int pPixelMeasurements[81] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,
    16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,35,36,
    37,38,39,41,42,43,45,46,48,49,50,51,53,54,55,56,58,59,60,61,
    62,63,64,65,66,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,
    83,84,85,86,87,88};
int *pCalibratedMeasurements = 0;
short lookupTable[256];
int i;
int pKernel2[9] = {0,1,0,1,1,1,0,1,0};
StructuringElement structElem2;
int pPixelMeasurements1[81] = {0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,
    16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,35,36,
    37,38,39,41,42,43,45,46,48,49,50,51,53,54,55,56,58,59,60,61,
    62,63,64,65,66,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,
    83,84,85,86,87,88};
int *pCalibratedMeasurements1 = 0;

// Initializes internal data (buffers and array of points for caliper measurements)
VisionErrChk(ivaData = IVA_InitData(12, 0));

// Creates a new, empty region of interest.
VisionErrChk(roi = imaqCreateROI());

// Creates a new line ROI contour and adds the line to the provided ROI.
VisionErrChk(imaqAddLineContour(roi, imaqMakePoint(784, 1232),

```

```

        imaqMakePoint(1148, 1488));

//-----//
//          Line Profile Grayscale          //
//-----//

VisionErrChk(IVA_LineProfile_Color(image, roi, IMAQ_RGB));

// Cleans up resources associated with the object
imaqDispose(roi);

VisionErrChk(IVA_CLRExtractValue(image));

//-----//
//          Filters: Edge Detection          //
//-----//

// Applies a nonlinear filter to highlight edges.
VisionErrChk(imaqEdgeFilter(image, image, IMAQ_EDGE_PREWITT, NULL));

//-----//
//          Filters: Convolution             //
//-----//

// Applies a linear filter to an image by convolving the image with a filtering kernel.
VisionErrChk(imaqConvolve(image, image, kernel, 3, 3, 0, NULL));

//-----//
//          Local Threshold                  //
//-----//

```

```
// Thresholds the image.
VisionErrChk(imaqLocalThreshold(image, image, 32, 32,
IMAQ_BACKGROUND_CORRECTION, 0.2, IMAQ_DARK_OBJECTS, 1));
```

```
//-----//
//          Basic Morphology          //
//-----//
```

```
// Sets the structuring element.
```

```
structElem.matrixCols = 3;
structElem.matrixRows = 3;
structElem.hexa = FALSE;
structElem.kernel = pKernel;
```

```
// Applies a morphological transformation to the binary image.
```

```
VisionErrChk(imaqMorphology(image, image, IMAQ_DILATE, &structElem));
```

```
//-----//
//          Basic Morphology          //
//-----//
```

```
// Sets the structuring element.
```

```
structElem1.matrixCols = 3;
structElem1.matrixRows = 3;
structElem1.hexa = FALSE;
structElem1.kernel = pKernel1;
```

```
// Applies a morphological transformation to the binary image.
```

```
VisionErrChk(imaqMorphology(image, image, IMAQ_DILATE, &structElem1));
```

```

VisionErrChk(IVA_Particle(image, TRUE, pPixelMeasurements, 81,
                          pCalibratedMeasurements, 0, ivaData, 7));

//-----//
//          Invert Binary Image          //
//-----//

lookupTable[0] = 1;

for (i = 1 ; i < 256 ; i++)
    lookupTable[i] = 0;

// Inverses the binary image.
VisionErrChk(imaqLookup(image, image, lookupTable, NULL));

//-----//
//          Basic Morphology          //
//-----//

// Sets the structuring element.
structElem2.matrixCols = 3;
structElem2.matrixRows = 3;
structElem2.hexa = FALSE;
structElem2.kernel = pKernel2;

// Applies a morphological transformation to the binary image.
VisionErrChk(imaqMorphology(image, image, IMAQ_OPEN, &structElem2));

//-----//

```



```

//          Advanced Morphology: Label          //
//-----//

// Labels the particles in a binary image by applying a unique value to all pixels
within a particle.
VisionErrChk(imaqLabel2(image, image, TRUE, NULL));

VisionErrChk(IVA_Particle(image, TRUE, pPixelMeasurements1, 81,
pCalibratedMeasurements1, 0, ivaData, 11));

// Releases the memory allocated in the IVA_Data structure.
IVA_DisposeData(ivaData);

Error:
    return success;
}

////////////////////////////////////
//
// Function Name: IVA_LineProfile_Color
//
// Description : Calculates the profile of the pixels along the boundary
//              of an ROI descriptor
//
// Parameters  : image - Input image
//              roi   - Region of interest
//              mode  - Color mode
//
// Return Value : success
//

```

//

```
static int IVA_LineProfile_Color(Image* image,
                                ROI*  roi,
                                int  mode)
{
    int success = 1;
    Image* tempImage;
    ROIProfile* roiPlane1Profile = NULL;
    ROIProfile* roiPlane2Profile = NULL;
    ROIProfile* roiPlane3Profile = NULL;

    //-----//
    //          Line Profile Color          //
    //-----//

    // Creates a temporary image mask.
    VisionErrChk(tempImage = imaqCreateImage(IMAQ_IMAGE_U8, 7));

    // Extracts the first color plane.
    VisionErrChk(imaqExtractColorPlanes (image, mode, tempImage, NULL, NULL));

    // Calculates the profile of the pixels along the boundary of an ROI descriptor.
    roiPlane1Profile = imaqROIProfile (tempImage, roi);

    // Extracts the second color plane.
    VisionErrChk(imaqExtractColorPlanes (image, mode, NULL, tempImage, NULL));

    // Calculates the profile of the pixels along the boundary of an ROI descriptor.
```

```
roiPlane2Profile = imaqROIProfile (tempImage, roi);
```

```
// Extracts the third color plane.
```

```
VisionErrChk(imaqExtractColorPlanes (image, mode, NULL, NULL, tempImage));
```

```
// Calculates the profile of the pixels along the boundary of an ROI descriptor.
```

```
roiPlane3Profile = imaqROIProfile (tempImage, roi);
```

Error:

```
imaqDispose(tempImage);
```

```
imaqDispose(roiPlane1Profile);
```

```
imaqDispose(roiPlane2Profile);
```

```
imaqDispose(roiPlane3Profile);
```

```
return success;
```

```
}
```

```
////////////////////////////////////////////////////////////////
```

```
//
```

```
// Function Name: IVA_CLRExtractValue
```

```
//
```

```
// Description : Extracts the value plane from a color image.
```

```
//
```

```
// Parameters : image - Input image
```

```
//
```

```
// Return Value : success
```

```
//
```

```
////////////////////////////////////////////////////////////////
```

```
static int IVA_CLRExtractValue(Image* image)
```

```

{
    int success = 1;
    Image* plane;

    //-----//
    //          Extract Color Plane          //
    //-----//

    // Creates an 8 bit image that contains the extracted plane.
    VisionErrChk(plane = imaqCreateImage(IMAQ_IMAGE_U8, 7));

    // Extracts the value plane
    VisionErrChk(imaqExtractColorPlanes(image, IMAQ_HSV, NULL, NULL, plane));

    // Copies the color plane in the main image.
    VisionErrChk(imaqDuplicate(image, plane));

Error:
    imaqDispose(plane);

    return success;
}

////////////////////////////////////
//
// Function Name: IVA_Particle
//
// Description : Computes the number of particles detected in a binary image and

```

```

//          a 2D array of requested measurements about the particle.
//
// Parameters : image          - Input image
//          connectivity        - Set this parameter to 1 to use
//                                connectivity-8 to determine
//                                whether particles are touching.
//                                Set this parameter to 0 to use
//                                connectivity-4 to determine
//                                whether particles are touching.
//          pixelMeasurements    - Array of measurement parameters
//          numPixelMeasurements - Number of elements in the array
//          calibratedMeasurements - Array of measurement parameters
//          numCalibratedMeasurements - Number of elements in the array
//          ivaData              - Internal Data structure
//          stepIndex            - Step index (index at which to store
//                                the results in the results array)
//
// Return Value : success
//
////////////////////////////////////
static int IVA_Particle(Image* image,
                        int connectivity,
                        int pPixelMeasurements[],
                        int numPixelMeasurements,
                        int pCalibratedMeasurements[],
                        int numCalibratedMeasurements,
                        IVA_Data* ivaData,
                        int stepIndex)
{
    int success = 1;

```

```

int numParticles;
double* pixelMeasurements = NULL;
double* calibratedMeasurements = NULL;
unsigned int visionInfo;
IVA_Result* particleResults;
int i;
int j;
double centerOfMassX;
double centerOfMassY;

//-----//
//          Particle Analysis          //
//-----//

// Counts the number of particles in the image.
VisionErrChk(imaqCountParticles(image, connectivity, &numParticles));

// Allocate the arrays for the measurements.
pixelMeasurements = (double*)malloc(numParticles * numPixelMeasurements *
sizeof(double));
calibratedMeasurements = (double*)malloc(numParticles *
numCalibratedMeasurements * sizeof(double));

// Delete all the results of this step (from a previous iteration)
IVA_DisposeStepResults(ivaData, stepIndex);

// Check if the image is calibrated.
VisionErrChk(imaqGetVisionInfoTypes(image, &visionInfo));

```

```

// If the image is calibrated, we also need to log the calibrated position (x and y)
ivaData->stepResults[stepIndex].numResults      =      (visionInfo      &
IMAQ_VISIONINFO_CALIBRATION ?
                                numParticles * 4 + 1 : numParticles * 2 + 1);

ivaData->stepResults[stepIndex].results = malloc (sizeof(IVA_Result) * ivaData-
>stepResults[stepIndex].numResults);

particleResults = ivaData->stepResults[stepIndex].results;

#ifdef IVA_STORE_RESULT_NAMES
    sprintf(particleResults->resultName, "Object #");
#endif
particleResults->type = IVA_NUMERIC;
particleResults->resultVal.numVal = numParticles;
particleResults++;

for (i = 0 ; i < numParticles ; i++)
{
    // Computes the requested pixel measurements about the particle.
    for (j = 0 ; j < numPixelMeasurements ; j++)
    {
        VisionErrChk(imaqMeasureParticle(image, i, FALSE, pPixelMeasurements[j],
&pixelMeasurements[i*numPixelMeasurements + j]));
    }

    // Computes the requested calibrated measurements about the particle.
    for (j = 0 ; j < numCalibratedMeasurements ; j++)
    {

```

```

        VisionErrChk(imaqMeasureParticle(image, i, TRUE,
pCalibratedMeasurements[j],
&calibratedMeasurements[i*numCalibratedMeasurements + j]));
    }

    #if defined (IVA_STORE_RESULT_NAMES)
        sprintf(particleResults->resultName, "Particle %d.X Position (Pix.)", i + 1);
    #endif
    particleResults->type = IVA_NUMERIC;
    VisionErrChk(imaqMeasureParticle(image, i, FALSE,
IMAQ_MT_CENTER_OF_MASS_X, &centerOfMassX));
    particleResults->resultVal.numVal = centerOfMassX;
    particleResults++;

    #if defined (IVA_STORE_RESULT_NAMES)
        sprintf(particleResults->resultName, "Particle %d.Y Position (Pix.)", i + 1);
    #endif
    particleResults->type = IVA_NUMERIC;
    VisionErrChk(imaqMeasureParticle(image, i, FALSE,
IMAQ_MT_CENTER_OF_MASS_Y, &centerOfMassY));
    particleResults->resultVal.numVal = centerOfMassY;
    particleResults++;

    if (visionInfo & IMAQ_VISIONINFO_CALIBRATION)
    {
        #if defined (IVA_STORE_RESULT_NAMES)
            sprintf(particleResults->resultName, "Particle %d.X Position (Calibrated)", i
+ 1);
        #endif
        particleResults->type = IVA_NUMERIC;

```



```

        VisionErrChk(imaqMeasureParticle(image, i, TRUE,
        IMAQ_MT_CENTER_OF_MASS_X, &centerOfMassX));
        particleResults->resultVal.numVal = centerOfMassX;
        particleResults++;

        #if defined (IVA_STORE_RESULT_NAMES)
            sprintf(particleResults->resultName, "Particle %d.Y Position (Calibrated)", i
+ 1);
        #endif
        particleResults->type = IVA_NUMERIC;
        VisionErrChk(imaqMeasureParticle(image, i, TRUE,
        IMAQ_MT_CENTER_OF_MASS_Y, &centerOfMassY));
        particleResults->resultVal.numVal = centerOfMassY;
        particleResults++;
    }
}

```

Error:

```

    free(pixelMeasurements);
    free(calibratedMeasurements);

    return success;
}

////////////////////////////////////
//
// Function Name: IVA_InitData
//
// Description : Initializes data for buffer management and results.

```

```

//
// Parameters : # of steps
//           # of coordinate systems
//
// Return Value : success
//
////////////////////////////////////
static IVA_Data* IVA_InitData(int numSteps, int numCoordSys)
{
    int success = 1;
    IVA_Data* ivaData = NULL;
    int i;

    // Allocate the data structure.
    VisionErrChk(ivaData = (IVA_Data*)malloc(sizeof (IVA_Data)));

    // Initializes the image pointers to NULL.
    for (i = 0 ; i < IVA_MAX_BUFFERS ; i++)
        ivaData->buffers[i] = NULL;

    // Initializes the steo results array to numSteps elements.
    ivaData->numSteps = numSteps;

    ivaData->stepResults = (IVA_StepResults*)malloc(ivaData->numSteps *
    sizeof(IVA_StepResults));
    for (i = 0 ; i < numSteps ; i++)
    {
        #if defined (IVA_STORE_RESULT_NAMES)
            sprintf(ivaData->stepResults[i].stepName, "");

```

```

#endif

ivaData->stepResults[i].numResults = 0;
ivaData->stepResults[i].results = NULL;
}

// Create the coordinate systems
ivaData->baseCoordinateSystems = NULL;
ivaData->MeasurementSystems = NULL;
if (numCoordSys)
{
    ivaData->baseCoordinateSystems =
(CoordinateSystem*)malloc(sizeof(CoordinateSystem) * numCoordSys);
    ivaData->MeasurementSystems =
(CoordinateSystem*)malloc(sizeof(CoordinateSystem) * numCoordSys);
}

ivaData->numCoordSys = numCoordSys;

```

Error:

```

return ivaData;
}

```

```

////////////////////////////////////

```

```

//

```

```

// Function Name: IVA_DisposeData

```

```

//

```

```

// Description : Releases the memory allocated in the IVA_Data structure

```

```

//

```

```

// Parameters : ivaData - Internal data structure

```

```

//
// Return Value : success
//
////////////////////////////////////
static int IVA_DisposeData(IVA_Data* ivaData)
{
    int i;

    // Releases the memory allocated for the image buffers.
    for (i = 0 ; i < IVA_MAX_BUFFERS ; i++)
        imaqDispose(ivaData->buffers[i]);

    // Releases the memory allocated for the array of measurements.
    for (i = 0 ; i < ivaData->numSteps ; i++)
        IVA_DisposeStepResults(ivaData, i);

    free(ivaData->stepResults);

    // Dispose of coordinate systems
    if (ivaData->numCoordSys)
    {
        free(ivaData->baseCoordinateSystems);
        free(ivaData->MeasurementSystems);
    }

    free(ivaData);

    return TRUE;
}

```

```

////////////////////////////////////
//
// Function Name: IVA_DisposeStepResults
//
// Description : Dispose of the results of a specific step.
//
// Parameters : ivaData - Internal data structure
//              stepIndex - step index
//
// Return Value : success
//
////////////////////////////////////
static int IVA_DisposeStepResults(IVA_Data* ivaData, int stepIndex)
{
    int i;

    for (i = 0 ; i < ivaData->stepResults[stepIndex].numResults ; i++)
    {
        if (ivaData->stepResults[stepIndex].results[i].type == IVA_STRING)
            free(ivaData->stepResults[stepIndex].results[i].resultVal.strVal);
    }

    free(ivaData->stepResults[stepIndex].results);

    return TRUE;
}

```

