

Министерство образования и науки Российской Федерации  
Московский физико-технический институт (государственный  
университет) Факультет управления и прикладной математики  
Вычислительный центр им. А. А. Дородницына РАН Кафедра  
«Интеллектуальные системы»

Фейзханов Рустем Усманович

Распознавание рукописной правки на изображениях  
документов в реальном времени в системе компьютерного  
зрения

010656 — Математические и информационные технологии

Выпускная квалификационная работа магистра

Научный руководитель:  
д. т. н. Местецкий Леонид Моисеевич

## Содержание

Аннотация .....	3
Введение .....	4
Литературный обзор .....	5
Теория .....	21
Преобразование Хаффа .....	21
Детектор ребер Кэнни .....	22
Гомография .....	25
Алгоритм .....	27
Бинаризация изображения .....	29
Трассировка изображений .....	30
Обнаружение ребер .....	31
Кластеризация ребер .....	32
Объединение кандидатов .....	33
Получение точек пересечения линий .....	33
Проведение гомографии .....	35
Обнаружение признаков .....	35
Соответствие признаков .....	37
Фильтрация признаков .....	38
Экстраполяция поля .....	39
Трансформация изображения .....	40
Вычитание изображений .....	41
Бинаризация изображения .....	42
Трассировка изображения .....	43
Реализация с помощью Javascript .....	44
Мотивация .....	44
Алгоритм .....	45
Пользовательское тестирование .....	46
Результаты .....	47
Заключение .....	48
Ссылки .....	49

## Аннотация

Задача распознавания краев прямоугольных структур достаточно известна. Тем не менее, почти все алгоритмы работают со статическими изображениями и не имеют ограничений на время выполнения. В данной работе рассматривается использование гомографии в реальном времени для видеопотока, который может быть получен с веб-камеры ноутбука. Кроме того, построен алгоритм, который может быть успешно использован для этого вида задач. Также полученный поток был использован для распознавания рукописных отметок на печатном документе. Основная цель состояла в том, чтобы создать программу, которая может быть использована для сканирования документа с помощью веб-камеры ноутбука с дополнительным распознаванием пометок. Главная задача была разработать алгоритм, который будет работать быстро (потому что требуется проведение обработки в реальном времени), а также с надлежащим качеством. Вторая задача заключалась в реализации и тестировании алгоритма, и конечная цель была создать программу для пользователя. Программа может найти документ на видео, затем найти рукописные пометки на документе с печатного текста, и затем классифицировать их на основе корректура знаки требованиям. Наконец, программа может положить оцифрованные пометки на цифровую версию документа.

## Введение

В настоящее время мультимедийные приложения реального времени (такие, как Skype, Viber, GTalk, MSN, ICQ и т.д.) становятся все более и более популярными. Кроме того, качество веб-камеры, резко возрастает. И люди используют все эти современные технологии, чтобы сделать электронные коммуникации ближе к реальным. Однако текущие реализации далеки от совершенства, и есть проблемы, которые остаются нерешенными. Одной из них является случай, когда люди должны показать их собеседнику картинку или документ на бумажной основе. Передать его в электронном формате может быть достаточно неудобно (например, из-за отсутствия сканера, или необходимости показать документ и изменения на нем в режиме реального времени). Это, в основном возникает, когда люди пытаются что-то дистанционно объяснить, нарисовать или написать на бумаге. Способы, используемые в настоящее время не являются эффективными и удобными (например, просто показать документ перед веб-камерой). Качество таких фотографий может быть низким из-за нескольких факторов:

### 1. Человеческие факторы

- . (a) Тряска рук;
- . (b) Постоянные изменения в положении документа;
- . (c) Поворот документа по отношению к камере ноутбука.

### 2. Технологические факторы

- . (a) Освещение (неравномерный свет, отражение от бумаги и т.д.);
- . (b) Качество камеры;
- . (c) Качество документа.

В этом проекте я предложил метод, который позволяет конвертировать и передавать видеопоток таким образом, что собеседник может увидеть документ, как если бы он сидел рядом с пользователем. В разделе 2 я провел обзор существующих работ в этой области. В разделе 3 я написал алгоритм, который я использовал для решения проблемы. В разделе 4 я написал о тестировании пользователей. В разделе 5 я проанализировал результаты тестирования. В разделе 6 я описал мои результаты и следующие шаги.



## Литературный обзор

Вся задача поиска и классификации пометок состоит из трех частей, каждая из которых является отдельной проблемой в компьютерном зрении. Многие исследователи сталкивались с этими проблемами. Кроме того, возникают некоторые специфические проблемы, которые не упомянуты в каких-либо конкретных научных работах.

Первая часть алгоритма состоит в обнаружении прямоугольных структур на изображениях. Есть шесть основных подходов, которые ученые используют для решения такого рода проблем. Первый подход к этой задаче предполагает использование Марковских случайных полей (МСП) [1]. Это метод обнаружения прямоугольных структур направлен на работу с различными видами внешних свойств объектов, в том числе размер, ориентацию и цвет. Метод отличается от обычных методов обнаружения прямоугольных объектов. В обычных реализациях, предполагают пять параметров (расположение центра (два параметра), длину, ширину и ориентацию) произвольно ориентированного прямоугольника, чтобы найти прямоугольные объекты быстро и исполнив ограничение на положение. Основная идея алгоритма состоит в применении МСП модели, которая построена на обнаруженных отрезках в картер ребер изображения, чтобы маркировать определенные сегменты линий, которые лежат на границе прямоугольных объектов. Вначале алгоритм обнаружения края на основе дифференциальной геометрии получает края изображения карты и направление градиента каждого реберного пикселя. Потом реберные пиксели с аналогичной ориентацией связываются в прямые отрезки и группы соседних параллельных сегментов линий объединяются в один сегмент линии. Наконец, МСП модель используется для маркирования ребер, принадлежащих к различным прямоугольным объектам.

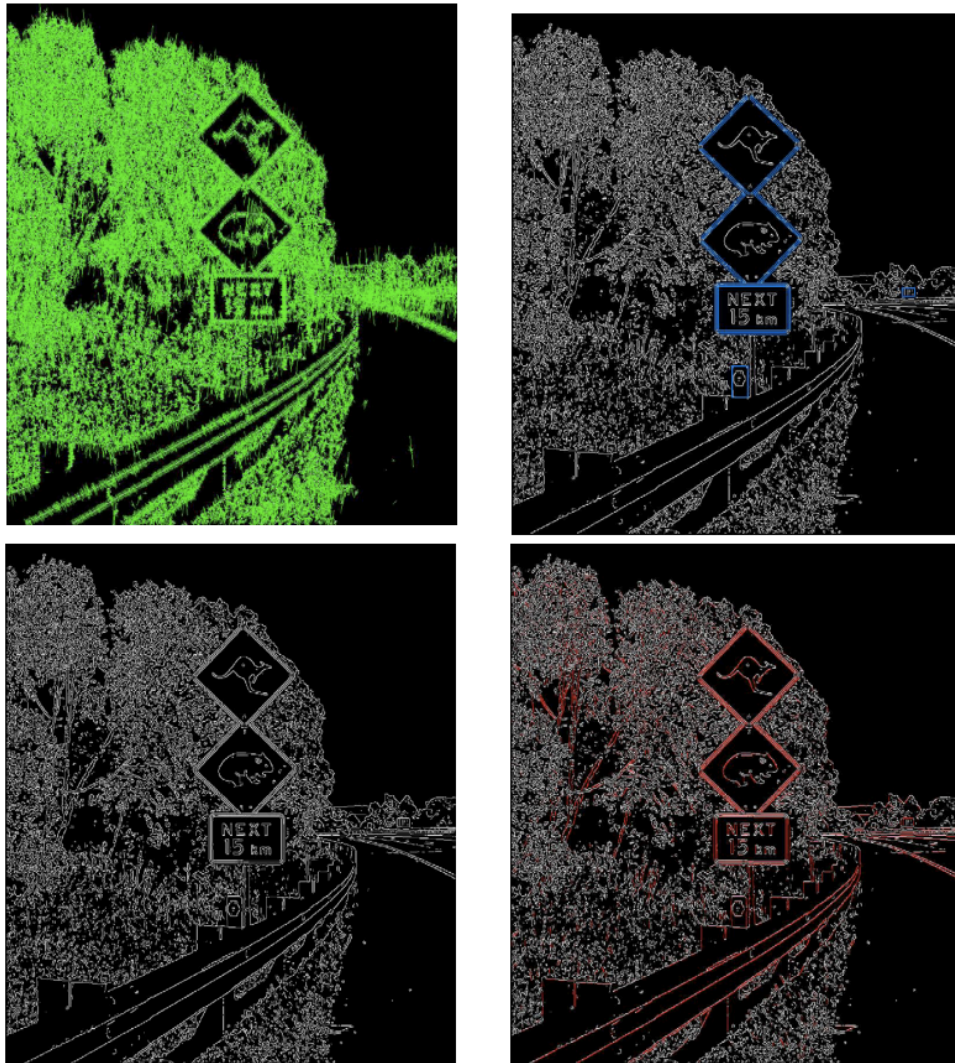


Fig.1 “MIP подход для нахождения прямоугольных структур”

Другой метод [2], который использует тот же подход, направлен на обнаружение полные или частичные прямоугольники и восстановление их позы в сложной широкой исходных условий. Способ также показывает, как прямоугольники могут служить в качестве опорных регионов для геометрической вычислений контекстом в случае плохо структурированных средах. Прямоугольник обнаружения формулируется как проблема маркировки в рамках ДПС, где основная структура график полученной обнаруженных отрезков, соединенных через ограничениями триангуляции Делоне. Этот подход позволяет выявить частичные прямоугольники, способствует обмену края между соседними прямоугольниками, и прямоугольник в прямоугольник конфигураций.

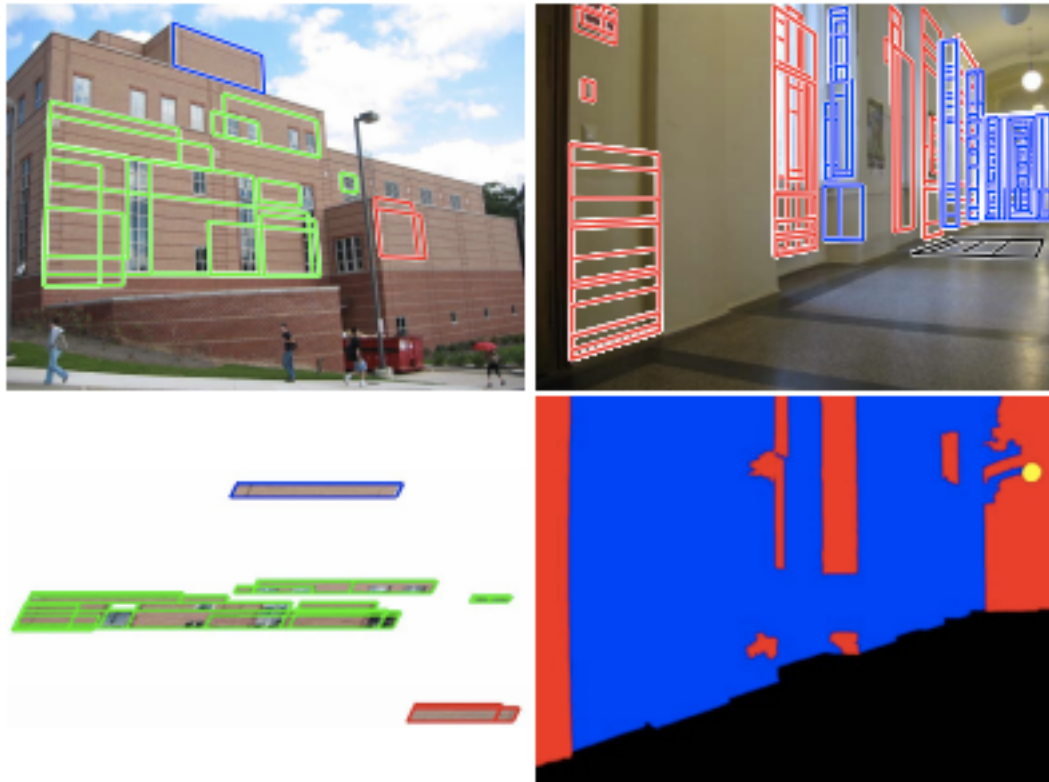


Fig.2 “МСП подход для зданий для нахождения прямоугольных структур”

Прямоугольные объекты могут быть обнаружены также используя метод, называемый "исчезающие линии" [3]. Этот детектор прямоугольников перспектива используется для поиска прямоугольники в проекции на плоскость изображения. Детектор обнаруживает вероятный 2D проекции прямоугольников в окружающей среде путем размещения четырехугольники со сторонами согласования с обнаруженными исчезающих линий в изображении. В то время как такие функции могут быть обнаружены с помощью методов, таких как обобщенной трансформатора Хью через восприятия группировки, время выполнения таких подходов делает их непригодными для приложений реального времени. Детектор перспектива прямоугольник обеспечивает точное обнаружение функций в течение операционного ограничения в режиме реального времени.

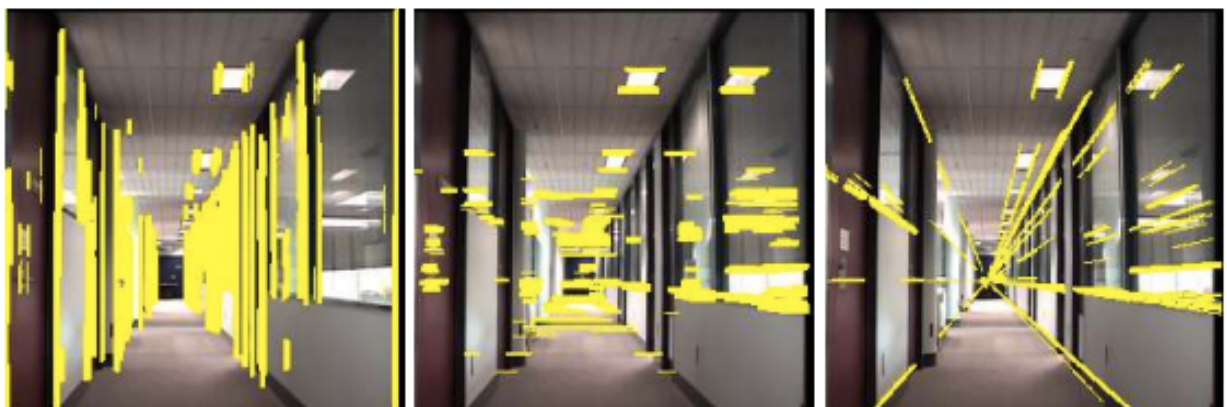


Fig.3 “Метод исчезающих линий”

Четвертый подход [4] называется метод параллельных линий. Этот метод используется для обнаружения и построения 3-D модели каркасные зданий с помощью несколько (два или более) панхроматические аэрофотоснимков. Изображение может быть принято в разное время и могут иметь различные решения. Предполагается, что здания являются прямолинейными (композиции прямоугольники) в форме и что крыши либо плоской (планарной и параллельно земле) или "симметричные" фронтоны (два симметричные наклонные поверхности). Гипотезу и проверить подход используется для построения обнаружение и описание. Гипотезы для прямоугольных секций крыши образуются путем группировки строк изображения в каждом представлении и проверка проводится путем объединения доказательства во всех доступных видов и ищет доказательства видимых стен и ожидаемых тени. Перцептивного процесс группирования используется для генерации гипотез представляет собой иерархическую один: линия пары сгруппированы для формирования параллели или L- переходы, которые затем используются, чтобы сформировать U-формы, которые, наконец, используемые для получения прямоугольных компонентов крыши. Особенности согласованы на каждом уровне иерархии. Таким образом, процессы группировки и согласования осуществляются одновременно.

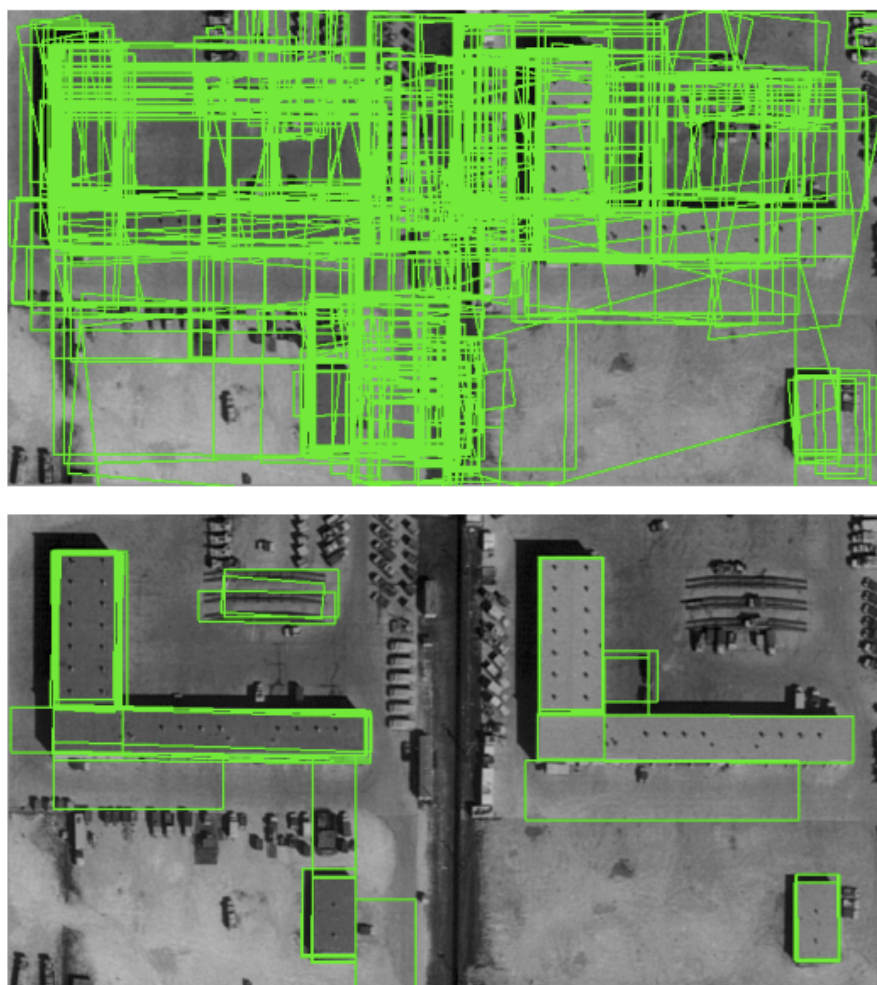


Fig.4 “Метод параллельных линий”



Одним из основных подходов [5] к проблеме нахождения прямоугольных структур на изображении оконный преобразование Хаффа. Этот метод использует геометрические характеристики прямоугольника в домене преобразования Хаффа, и такие характеристики используются для определения прямоугольника непосредственно в пространстве Хаффа. Этот метод работает для структуры прямоугольника с неизвестными размерами и ориентацией и не нужно группировку и извлечение линейных сегментов. Все пиксели изображения сканируются, и раздвижные окна метод используется для расчета преобразования Хаффа регионов изображения. Пики на изображении Хаффа позже выборе и прямоугольник обнаруживается, когда четыре извлеченные пики предоставить конкретные геометрические требования. Экспериментальные результаты показывают, что предложенный метод многообещающие результаты для синтетических и природных образов.

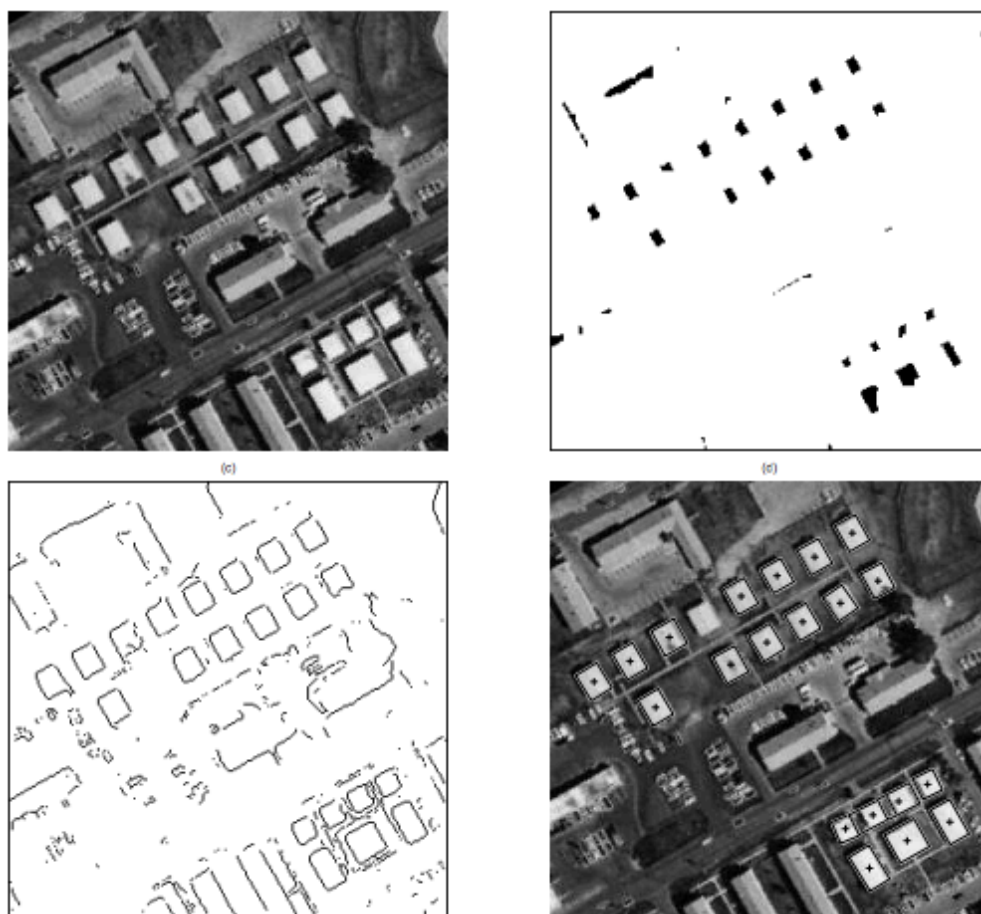


Fig.5 “Метод трансформации Хаффа с использование окна”

Широко распространен метод [6], использующий рекурсивный вызов метода. Решение зависит от того факта, что путем принятия соответствующих одномерных проекций данных проблема часто становится простым..

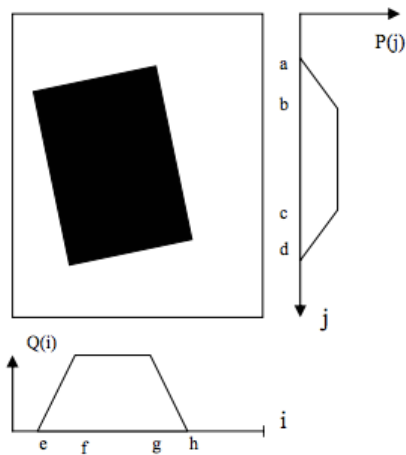


Fig.6 “Одномерный проекционный метод”

Наконец, некоторые исследователи используют метод дифференциальной омографии [7] найти прямоугольные структуры. Метод используется для выявления дорожки плоскости, используя подход, омографии основе. Пиксель-пара сопоставления характеристик локальных областей используется для оценки временной разницы омографии матрицы между двух последовательных видеокadres; регионы, которые удовлетворяют оценкам дифференциального омографии классифицируются как плоскости дороги. Дифференциальный омография оценивается оптических потоков дорожного плоских областей при использовании алгоритма RANSAC извлечь контрольные оптические потоки. Так дифференциального омография рассчитывается соотношение между изображением координировать и потоки в изображении предложенный алгоритм работает без требования оценки самого омографии матрицы. Именно поэтому предлагаемый алгоритм может быть применен без каких-либо дополнительной калибровки. Предложенный алгоритм эффективно обнаруживает оптические потоки от дороги региона с помощью функции соответствия пикселей пара.

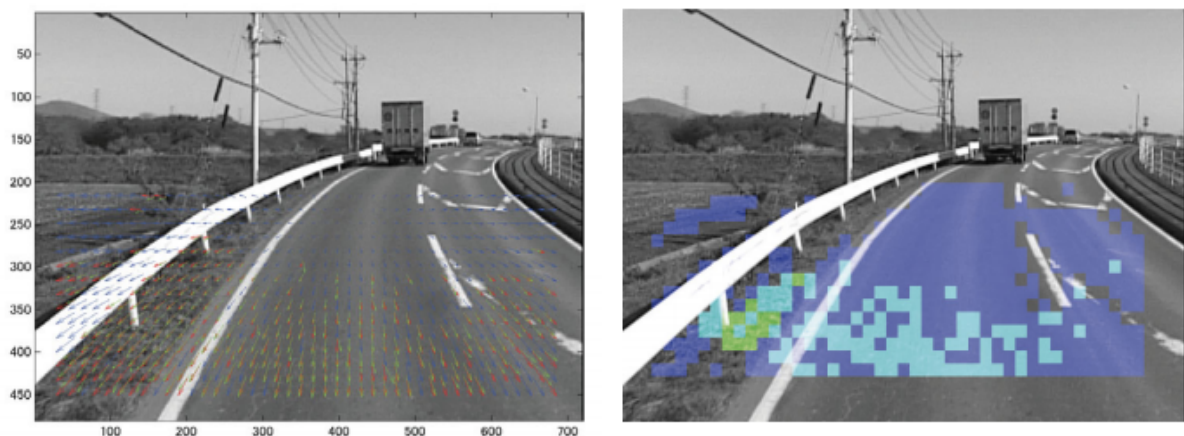


Fig.7 “Метод дифференциальной гомографии”

Вторая часть задачи является выявление изменений на основе серии изображений. Есть разные работы в этой области. Одним из основных и простейших подходов [8] является использование порога. Четыре различных метода выбираются при выборе пороговых значений, которые работают на очень разных принципах. Эти методы:

- 1) Нормальная модель используется для распределения интенсивности шума,
- 2) интенсивности сигнала проходят проверку решений местных сравнения распределения интенсивности в двух кадрах изображения (т.е. разница карта не используется),
- 3) пространственные свойства шума моделируются распределения Пуассона
- 4) пространственные свойства сигнала смоделированы в качестве стабильного количества областей (или стабильной числа Эйлера)

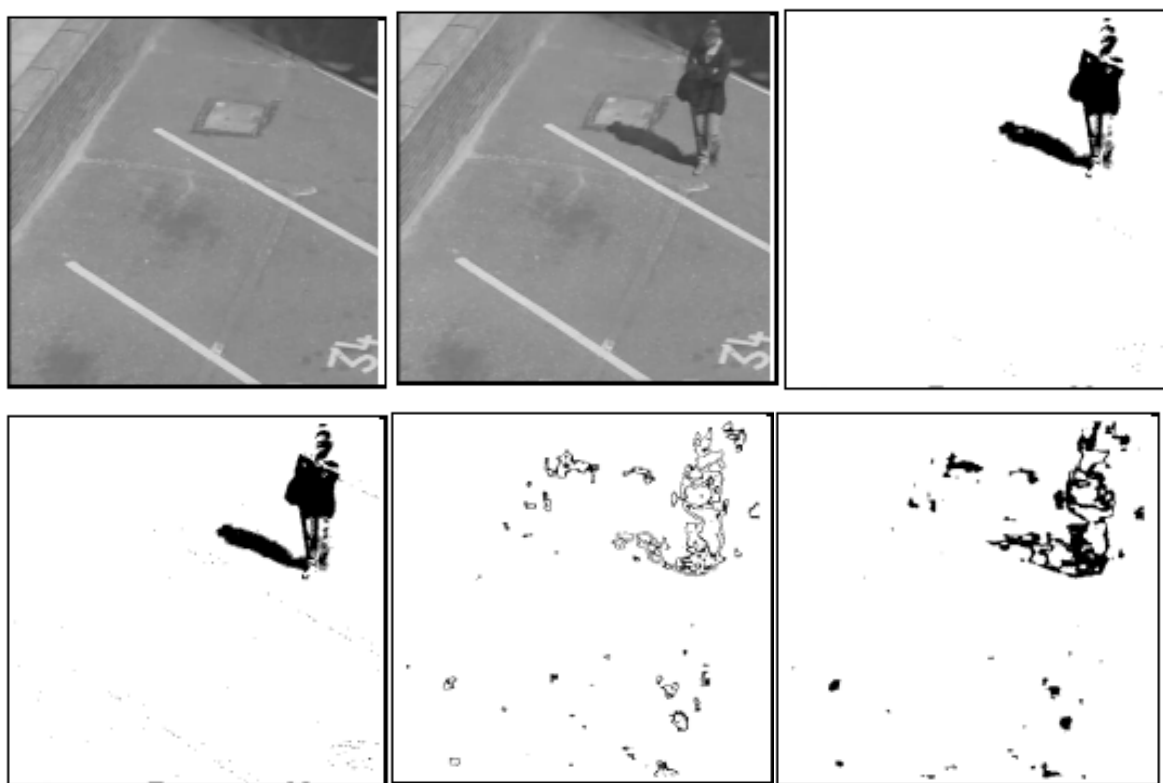


Fig.8 “Типы порогов и их влияние”

Третий подход [9], основан на консенсусе фоновых слов, который регулирует свои внутренние параметры с помощью обратной связи, чтобы выдерживать сложные условия, сохраняя при этом хороший уровень чувствительности. Придумал "PAWCS", ключевые полномочия этого метода заключаются в его очень упорной и сильной словаре модели, основанной на позиционных и бинарных функций цвета и на его способности к адаптации на уровне пикселей производительность сегментации. Эксперименты показывают, что он превосходит многие предлагаемые решения с точки зрения общей производительности. Кроме того, метод вводит в режиме реального времени обучения и регулирования

возможностей и дополнительный словарь уровня кадра. Его зависимость от местных двоичных подобия Patterns на уровне пикселей для обнаружения пространственно-временных различий и их динамически ограничена, обратная связь управляемых управления параметр также способствовать большим, чтобы его новизны. Точнее, вклады:

- 1) введение новой модели, которая может принимать и сохранять исходные данные в течение периодов времени. Он использует LBSP функции, значения цвета и показатели сохраняемости, которые сгруппированы в фоновых слов, чтобы получить пространственно-временных данных на уровне пикселей. Эта модель связывает местные и глобальные справочные представления в процессе двухступенчатой классификации.
- 2) предложение о управления с обратной связью, которая может принимать и обрабатывать кадр регионов, которые основаны на фоне динамики и результатов сегментации. Так как это дает возможность для внутренних значений, чтобы отрегулировать после классификации каждого пикселя, это, главным образом, устойчивым к шумных областей и динамического фона движения.

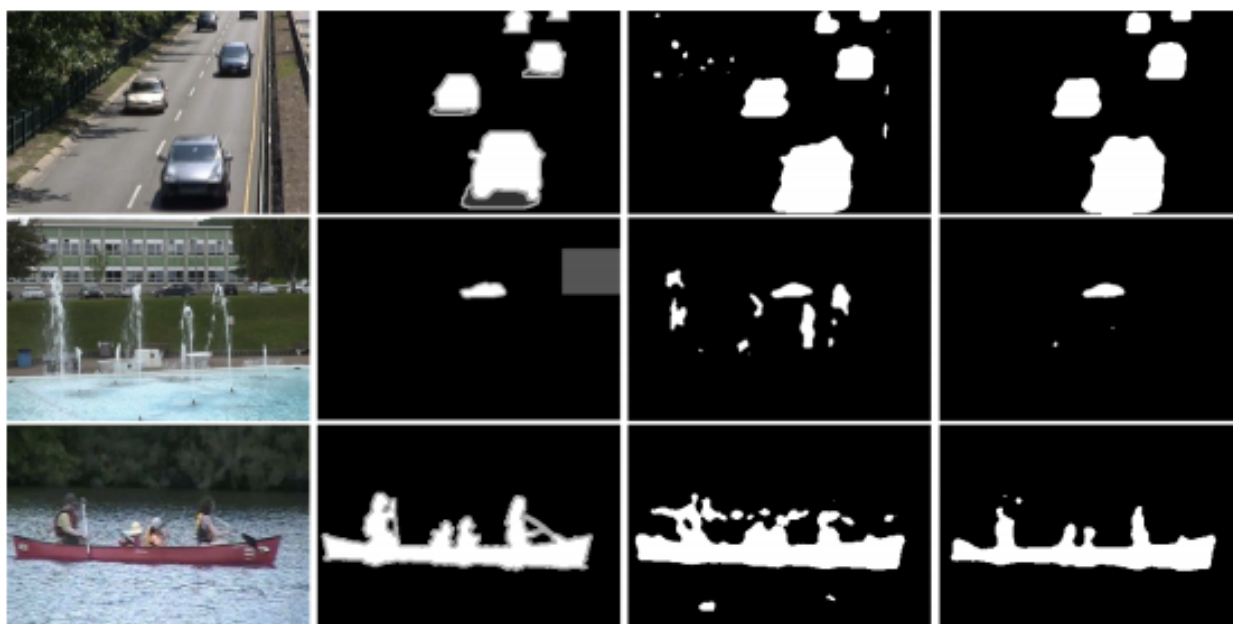


Fig.9 “Метод консенсуса фоновых слов”

Один из подходов [10] к задаче решает задачу о нахождении правильного метрику для обнаружения изменений. Процесс нахождения правильного метрику включает в себя определение и сравнение шесть разница изображение функции, в основном на основе RGB и HSV представлений изображений. Метрика для выявления изменений должны быть использованы для:

1. вариации равномерную яркость освещения (яркость источника света, камеры диафрагмы или чувствительности)



2. равномерное освещение спектральные вариации (спектральная чувствительность камеры или источник света спектрального состава)
3. единые локальные вариации яркости освещения (источники света или рядом тени)

По данным газеты глобальное освещение делает значительно не изменится, нормально евклидово расстояние в RGB или HSV изменены пространство дает хорошие результаты. Когда есть существенные изменения в спектре изображения, то HSV дает лучшие результаты. Наконец, если имеется большое локальное изменение освещенности, ни один из исследованных методов не работает хорошо, но компоненты H и S дают хорошие результаты.



Fig.10 “Пример выявления изменений для анализа различных метрик”

Существует очень интересный подход [11] к проблеме, которая называется автоматический анализ разности изображения. Эта методика обрабатывает два многоспектральные изображения, полученные на двух отдельных дат, чтобы генерировать дополнительную изображение. Рассчитывается разница изображение таким образом, что значения пикселей по сравнению с почвенно-растительного покрова изменения текущей стоимости резко отличаются от тех, из пикселей, связанных с устойчивыми областях. Изменения затем отличает изучения (т.е. порог) разностного изображения. Например, одномерный метод изображения генерирует разностный разностного изображения вычитанием одного спектрального диапазона двух мультиспектральных изображений. Выбор спектральной полосы зависит от конкретного типа изменения, которые будут обнаружены. Аналогичное понятие используется широко приняты изменения векторный анализ (ОНМК) техники. В этом примере несколько каналов спектра используются в каждый момент времени. Для каждой пары соответствующих пикселей, спектральный вектор изменения "рассчитывается как вариации между векторами признаков в два раза. Затем значения пикселей в изображении разности сравниваются с модулями спектральных векторов изменения. Отсюда следует, что пиксели, которые не изменились,

присутствующие небольшие значения, в то время как пиксели, которые изменили существующие, а большие значения.

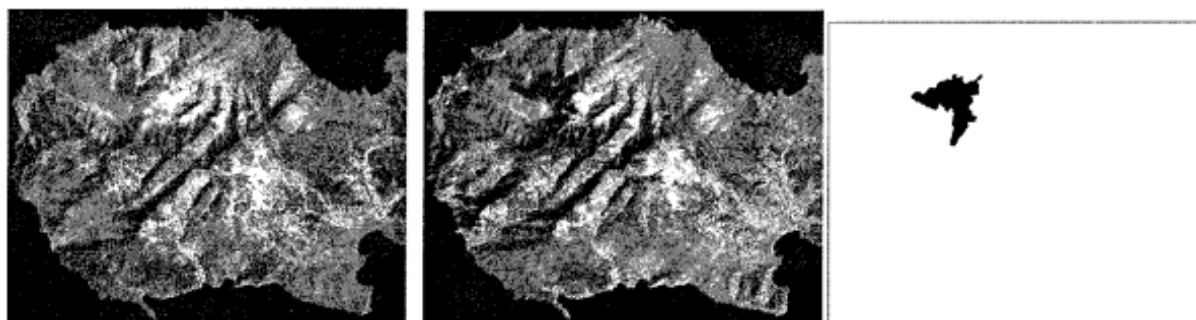


Fig.11 “Автоматический анализ методом разниц изображений”

Одно из интересных приложений использует нейронные сети подход [12] для обнаружения изменений в движении среды. Задача обнаружения изменений в окружающей среде вождения состоит из трех частей. Во-первых, так как некоторые подсистемы составляют систему помощи водителю, подсистемы должны быть организованы, чтобы достичь оптимального исполнения системы помощи водителю. Объединение подсистем, в том числе, которые, когда и как они будут использоваться, часто зависит от обстоятельств вождения условиях. Во-вторых, значения встроенных в подсистемах должны быть восстановлены изменений окружающей среды, таких как вариации в освещенности, погоды, дорожных условий, скорости и т.д., наконец, неожиданные различия в условиях вождения часто связанных с критических ситуаций трафика. Изменения, находящиеся на рассмотрении включают перемещение на полосу слева, двигаясь в переулок справа, вход шоссе, шоссе выхода, въезда в туннель, выходу из туннеля, и преодолеть вперед. Чтобы определить вышеуказанные изменения в движении среды, вычислительная база приводится в когнитивной обработки человеческого и избирательное внимание предлагается. В рамках этого, оба пространственные и временные данные извлекаются из видео последовательностей. Извлеченные данные служит в качестве входных стимулов к пространственно-временной внимание (СТА) нейронной сети. Если стимулы держать иннервирующих нейронной сети, внимание будет создана в сети. Внимание картина связана с акцентом, вместе с направлением и места движения фигуры, сделать то, что называется категорический особенность. Исходя из этого признака группового рисунка внимания и, в свою очередь, изменения в движении среды, соответствующей группы, определяются с помощью конфигурируемый адаптивный теорию резонанса (АТР) нейронной сети.

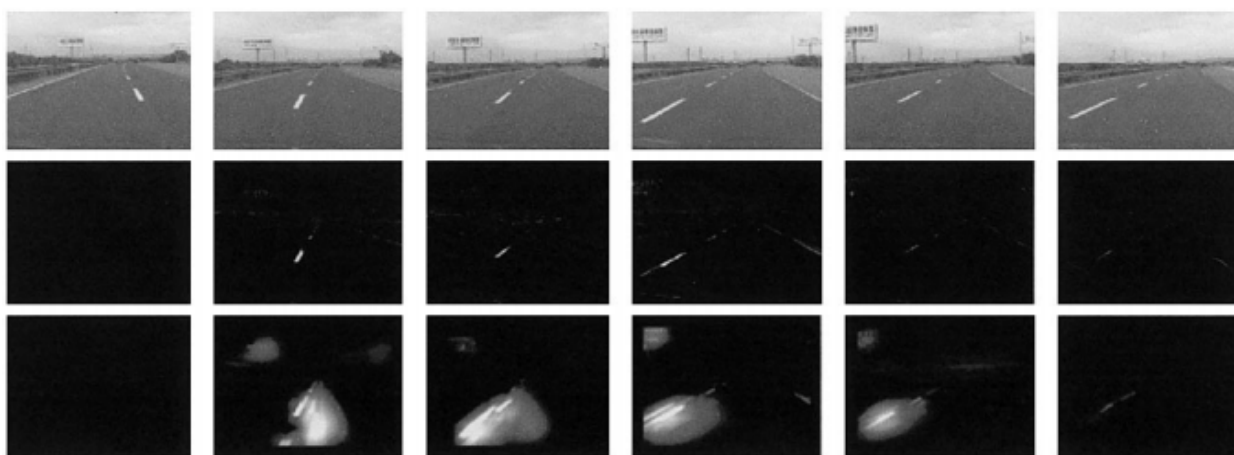


Fig.12 “Использование нейронных сетей для анализа окружающей среды вождения”

Есть много различных методов [13] для обнаружения изменений, упомянутых в статье. Цель алгоритма обнаружения изменения для обнаружения "значительные" изменения, отвергая "неважных" те. Сложные способы получения этой различие требует детального моделирования всех ожидаемых типов изменений (важных и неважных) для данного приложения, и интеграции этих моделей в качестве эффективного алгоритма. В следующих подразделах описываются шаги предварительной обработки, используемые для подавления или отфильтровать распространенные типы "неважных" изменений до принятия решения обнаружения изменений. Эти шаги включают в себя геометрические и радиометрические коррективы.

Заключительная часть задачи является классификация почерк марок. Есть разные подходы, основанные на конкретных приложений. Для применения [13] признания доске отмечает новая система вводится. Примечания, написанные на доске можно рассматривать в качестве нового механизма в исследовании распознавания рукописного, который получил небольшую внимание в прошлом. Для признания офф-лайн СММ-распознаватель используется. Распознавания завершается методов обработки информации он-лайн и генерации изображений. Схема состоит из шести основных частей: он-лайн предварительной обработки, преобразования информации офф-лайн, офф-лайн предварительной обработки, выделения признаков, классификации и пост-обработки. Скорость признание распознавания в независимом эксперименте. Применяя государством в самых современных методов, таких как оптимизация количества государств и гауссовых компонент, и в том числе языковой модели значительное увеличение может быть достигнуто.



In mid-april, Anglosey  
moved his family and  
entourage from Rome to Naples,  
here to await the arrival of

In mid-april Anglosey  
moved his family and  
entourage from Rome to Naples,  
here to await the arrival of

Fig.13 “Система распознавания пометок на доске”

Второй подход [14] включает в себя использование нейронных сетей. В большинстве систем, эти два элемента обрабатываются отдельно, со сложными методами предварительной обработки, используемых для извлечения особенности изображения и последовательные модели, такие как ПММ, которые используются для обеспечения транскрипции. Объединив два последних инноваций в нейронных сетях, многомерных периодических нейронных сетей и коннекционистского временной классификации - в глобальном масштабе подготовку форума распознавания рукописного текста вводится, который принимает сырую информацию пикселей в качестве входных данных. По сравнению с конкурирующими системами, она не требует какой-либо конкретной алфавит предварительную обработку, и может быть использован таким же образом, для любого типа языка. Проблема остается, по-прежнему, как превратить двухмерные изображения в одномерных последовательностей меток. Решение передать информацию через иерархию MDRNN частей, блоков, собранных вместе активаций после каждого уровня. Высоты блоков были выбраны, чтобы постепенно свернуть 2D изображения на 1D последовательностей, которые могут быть помечены в выходном слое. Такие иерархические структуры очень часто используется в компьютерном зрении, потому что они позволяют сложные функции должны быть построены до поэтапно. В частности, многослойная структура аналогична той, которая используется свертки сетей, хотя следует отметить, что из-за свертки сети не рецидивирующий, они не могут быть использованы для распознавания рукописного текста рукописным без предварительного сегментированных входов.

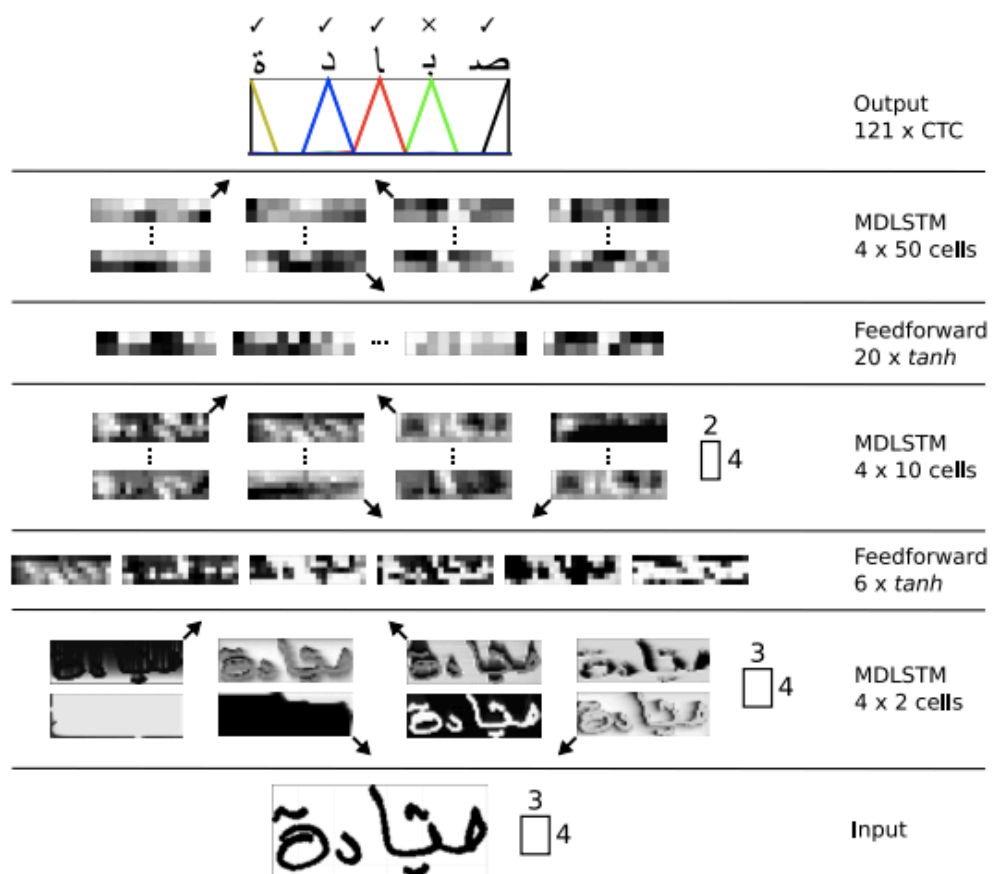


Fig.14 “Использование нейронных сетей для распознавания рукописного текста”

Третий подход [15], используя генетические алгоритмы. Основная идея генетического алгоритма идет из того, что он может быть использован как превосходный помощник для комбинирования различных стилей написания характер и генерации новых стилей. Внимательно рассматривая возможность человеческого разума в признании почерка, он может быть обнаружено, что люди могут распознавать символы, даже если они, видя, что стиль впервые. Это может быть достигнуто из-за их власти, чтобы визуализировать части стилей в неизвестном характера. Основной принцип, чтобы положить такую же мощь в машинах. Первый этап процесса собирает множество изображений символов. Второй шаг заключается в преобразовании их в графиках. Графики всех персонажей были перемешаны, чтобы создать стили промежуточные между стилями родительские символов распознавания символов, включенных в соответствие полученного графика с графиками, которые создаются путем смешивания.

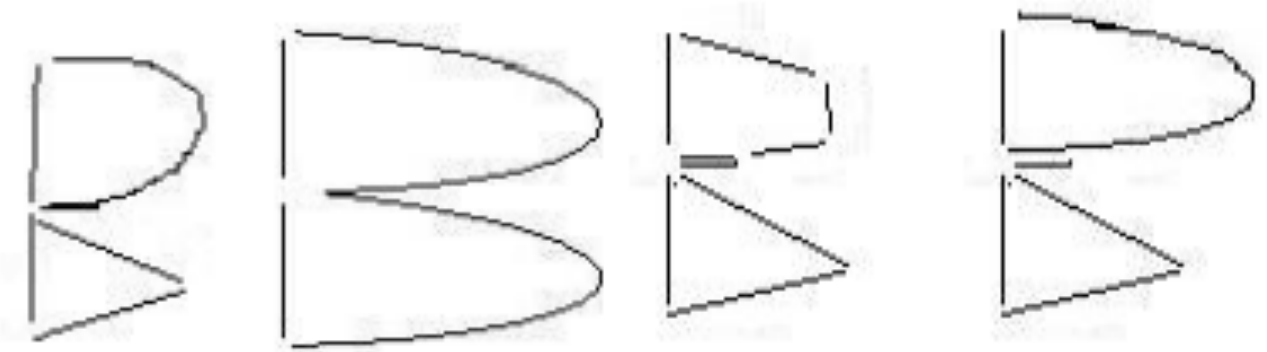


Fig.15 “Метод генетических алгоритмов”

Один из самых интересных подходов [16] для распознавания рукописного текста является рецидивирующий нейронные сети подход, который используется дико. Система основана на принципе способен непосредственно расшифровки необработанные данные рукописного ввода. Система состоит из повторяющейся нейронной сети с выходного слоя, предназначенного для маркировки потока, в сочетании с вероятностной модели языка. В экспериментах на неограниченной базой данных, система дает хорошие результаты, используя как сырые или предварительно обработанных данных, опережая государство-оф-искусство систему СММ в обеих ситуациях. Архитектура РНН является двунаправленным Длинные кратковременной памяти, которая выбрана для его способность обрабатывать данные с длинными зависимостей времени. РНН использует недавно представила Коннекционистский временную выходной классификация слой, который был специально разработан для маркировки данных несегментированной последовательности. Алгоритм вводится для применения ограничений к сети выходов, тем самым обеспечивая транскрипции на уровне слов.

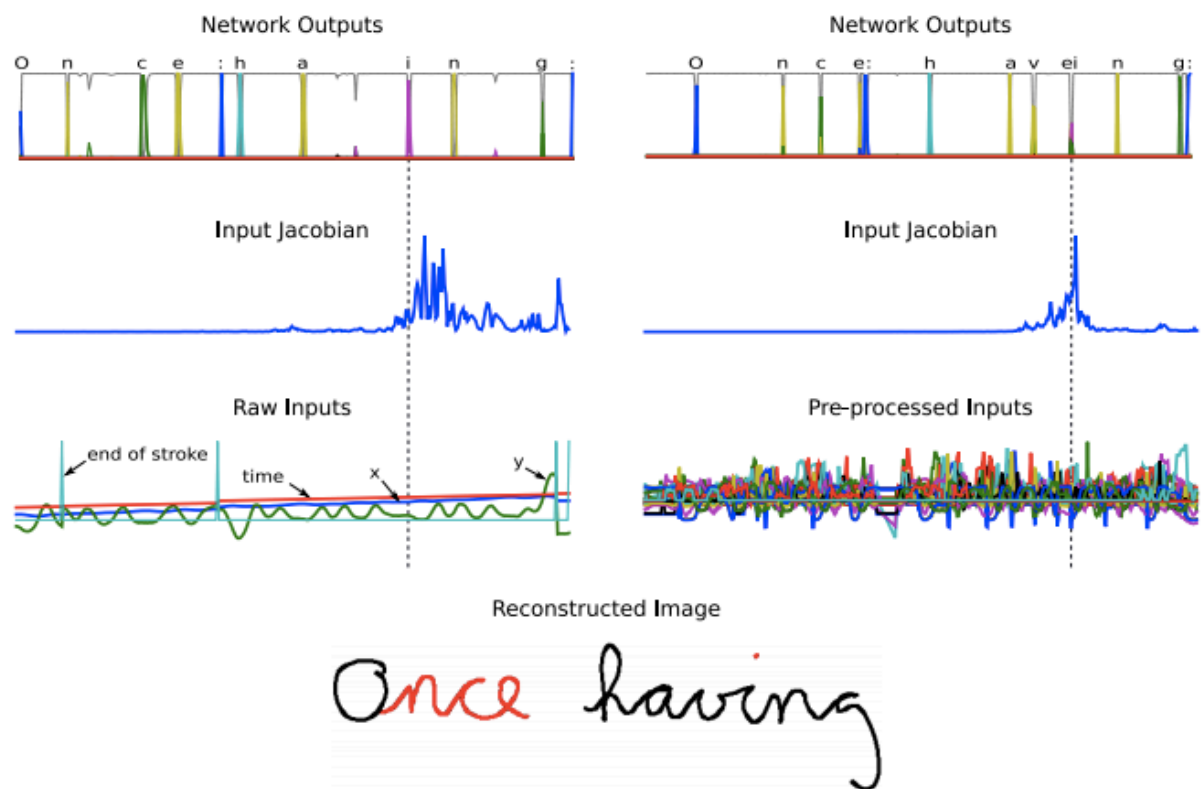


Fig.16 “Метод периодических нейронных сетей”

Наконец, для онлайн-распознавания рукописного текста есть распространенный метод [17], который использует метод опорных векторов. Этот метод сочетает в себе динамические временного масштаба (DTW) и поддерживает векторные машины (SVM) путем создания нового ядра SVM. Этот тип ядра называется гауссовой DTW (GDTW) ядра. Такой подход ядра имеет преимущество над другими методами ТММ. Она не учитывает модель для условных плотностей класса. Вместо этого, он обращается к проблеме дискриминации путем создания пределы класса и, следовательно, менее чувствительны к допущений моделирования. С помощью DTW в функции ядра, общие проблемы классификации, связанные с переменными размера последовательных данных может быть обработан.

$T$	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$
$\phi^*$					
$D(T, P_j)$	0	0.20	0.71	0.99	10.04
$K(T, P_j)$	1	0.70	0.28	0.17	0.00

Fig.17 “Метод SVM”

Чтобы решить задачу я использую сочетание различных методов. Основная цель заключается в преобразовании исходного изображения в соответствующей форме. Чтобы сделать это, я применяю омографии против исходного изображения. Сначала я обнаруживаю все края в изображении, используя хитрые детектор края, то я обрабатываю края, полученные, и, наконец, получить углы бумаги так, что я могу провести омографии. После этого, я буду иметь изображения серии, где мне нужно, чтобы найти изменения. Для этого я бегу RANSAC и точки экстраполировать смещения, которые в области сдвига конца продукции, что помогает вычесть одно изображение от другого без решения существующих текст на нем. Наконец, я Vinarize вычитание и обнаружить следы. После этого, я могу классифицировать их с использованием стандартных методов распознавания текста почерк.



# Теория

## Преобразование Хаффа

Одной из основных проблем в области компьютерного зрения [18], которая часто возникает является обнаружение простых форм, например, прямых линий, эллипсов или кругов. Как правило, детектор ребер используется в качестве предобработки, чтобы получить пиксели и точки, которые находятся на кривой в изображении. Из-за ошибок в информации об изображении или детекторе ребер могут отсутствовать точки и пиксели на кривых и другие ошибки, такие как отклонения между кривой и точками, так как они получены от детектора ребер. Из-за этого часто сложно организовать признаки ребер к соответствующему набору форм. Основная цель преобразования Хаффа состоит в том, чтобы решить эту проблему, позволяя выполнять группирование точек в кандидаты формы путем проведения процедуры голосования по набору параметризованных графических объектов.

Одним из основных случаев преобразования Хаффа является обнаружение прямых линий. Прямые представлены как  $y = ax + b$  - точки в пространстве параметров. Тем не менее, вертикальные линии представляют собой проблему, потому что они делают один из параметров слишком большим. Именно поэтому было предложена следующая нормальная форма:

$$r = x \cos \theta + y \sin \theta$$

где  $r$  - расстояние от начала координат до линии, а  $\theta$  - угол между осью  $x$  и ближайшей точкой.

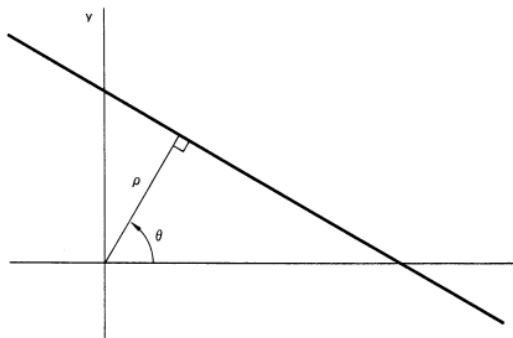


Fig.18 “Пример линии”

Именно поэтому мы можем связать пару  $(r, \theta)$  с любой линией на изображении. Плоскость пар называют пространством Хаффа для группы линий в двух измерениях.

Имея одну точку на плоскости, мы будем иметь набор всех линий, которые проходят через эту точку и соответствуют кривой в  $(r, \theta)$  пространстве, которое является уникальным для этой точки. Группа из двух или более точек, которые образуют прямую линию будет производить кривые, которые пересекаются на  $(r, \theta)$  для этой линии. Именно поэтому проблема поиска коллинеарных точек преобразуется в задачу обнаружения параллельных кривых.

## Детектор ребер Кэнни

Детектор ребер Кэнни [19,20] состоит из пяти этапов:

- 1) Сглаживание
- 2) Поиск градиентов интенсивности
- 3) Применение подавления не максимумов
- 4) Применение двойного порога
- 5) Получение ребер

Общие критерии для определения границ состоят из трех пунктов:

- 1) Низкий уровень ошибок (обнаружение должно найти столько ребер сколько возможно)
- 2) Локализация (точки ребра должны находиться в середине ребра)
- 3) Уникальность (ребро в изображении может быть зарегистрировано только один раз)

### Сглаживание

Мы используем фильтр Гаусса для удаления шума на изображении, потому что это очень важно, чтобы блокировать ложное обнаружение ребер. Фильтр сглаживания уменьшает воздействие шума на детектор ребер. Гауссова матрица ядра фильтра с размером ядра  $k$  выглядит следующим образом:

$$H_{ij} = \frac{1}{2\pi\sigma^2} * \exp\left(-\frac{(i-k-1)^2 + (j-k-1)^2}{2\sigma^2}\right)$$

Важно отметить, что размер гауссова ядра значительно влияет на производительность детектора. Чем больше размер, тем больше устойчив детектора к шуму. В дополнение к этому ошибка локализация детектора края будет возрастать с увеличением размера ядра

фильтра. Эксперименты показывают, что 5 является хорошим размером для большинства примеров, но это зависит от конкретной ситуации.

#### Поиск интенсивности градиента изображения

Алгоритм Кэнни применяет четыре фильтра для обнаружения горизонтальных, вертикальных и диагональных ребер в размытом изображении потому что ребра на изображении могут иметь разные направления. Методы обнаружения ребер, такие как Робертс, Превит и Собель позволяют получить первую производную в горизонтальном направлении и вертикальном направлении. Из этих значений градиент и направление ребра могут быть определены следующим образом:

$$G = \sqrt{G_x^2 + G_y^2}$$
$$\Theta = 2 \arctan \left( \frac{G_y}{G + G_x} \right)$$

где  $G$  может быть вычислен с помощью функции  $\text{sqrt}$  и  $\text{arctan}$ . Направление ребра может быть одним из четырех направлений: вертикальным, горизонтальным и двух диагоналей. Направления ребра для каждого цветового сегмента будут иметь определенные значения углов.

#### Применение подавления не максимумов

Подавления не максимумов используется для уменьшения толщины ребер. После расчета градиента, полученное ребро по-прежнему размыто. Чтобы следовать критерию уникальности, должен быть только одно значение края. Таким образом, не-максимальное подавление помогает обнулить значение все градиентов за исключением локального максимума, который имеет высокое значение изменения интенсивности. Алгоритм для каждого пикселя в градиентного изображения:

- 1) Сравнение значения ребра текущего пикселя с значениями пикселей, находящихся на положительном и отрицательном направлениях градиента пикселя.
- 2) Если это он больше по сравнению с другими пикселями в маске с теми же направлениями, то значение сохраняется. Иначе - будет подавлено.

Иногда алгоритм объединяет градиентные направления во множество дискретных направлений, и после этого использует фильтр  $3 \times 3$  на выходе из предыдущей части

алгоритма. В каждом пикселе, значение градиента ребра равно нулю, если ее величина меньше величины двух соседей в направлении градиента. Знак направления не имеет значения.

#### Применение двойного порога

После применения подавления, реберные пиксели соответствуют реальным ребрам. Тем не менее, есть еще некоторые реберные пиксели, вызванные шумом и изменением цвета. Для того, чтобы избавиться от ложных пикселей важно отфильтровать пиксели со слабым значением градиента и сохранить ребра с высоким значением градиента. Таким образом, два пороговых значения устанавливаются для разъяснения различных типов реберных пикселей, один называется высоким пороговым значением, а другой называется низким пороговым значением. Если значение градиента реберного пикселя выше, чем высокая пороговая величина, они помечаются как сильные реберные пиксели. Если значение градиента реберного пикселя является меньшим, чем высокое пороговое значение и больше, чем низкое пороговое значение, они помечаются как слабые реберные пиксели. Если значение пикселя меньше, чем низкое пороговое значение, они будут отфильтрованы. Два пороговых значения - эмпирически полученные значения, которые должны быть определены при работе с различными изображениями.

#### Получение ребер

Согласно критериям пиксели с высоким значением ребра будут включены в окончательные ребра изображения, потому что они соответствуют реальным ребрам изображения. Проблема остается с пикселями со слабым значением ребра, потому что эти пиксели могут соответствовать истинному краю, а могут и шуму и цветовым изменениям. Для достижения точного результата, необходимо удалить пиксели со слабым значением ребра, который появились из-за шума. Пиксели со слабым значением ребра соответствующие реальным ребрам будут связаны с пикселями с высоким значением ребра. BLOB-анализ используется, чтобы найти эту связь. Этот анализ основан на рассмотрении пикселя с низким значением ребра и 8-связной окрестности пикселя. Если есть один сильный край пиксель в BLOB, то пиксель остается. В противном случае, он может быть удален.

## Гомография

Мы можем связать два изображения одной и той же плоской поверхности в пространстве с помощью гомографии. После того, как вращение камеры и трансформация рассчитываются исходя из матрицы гомографии, можно провести гомографию.

Давайте рассмотрим камеру [19], которая направлена в точки  $P_i$ , которые расположены на плоскости. Зададим воображаемую камеру  $b$ , которая направлена перпендикулярно плоскости. Мы можем рассчитать проекции  $P_i$  из  $a$  в  $b$ :

$$p_i^b = K_b * H_{ab} * K_a^{-1} * p_i^a$$

где  $H_{ab}$

$$H_{ab} = R - \frac{tn^T}{d}$$

$R$  является матрицей вращения

$t$  вектор трансформации

$n$  и  $d$  являются нормальными векторами плоскости

$K_a$  и  $K_b$  матрицы параметров камер.

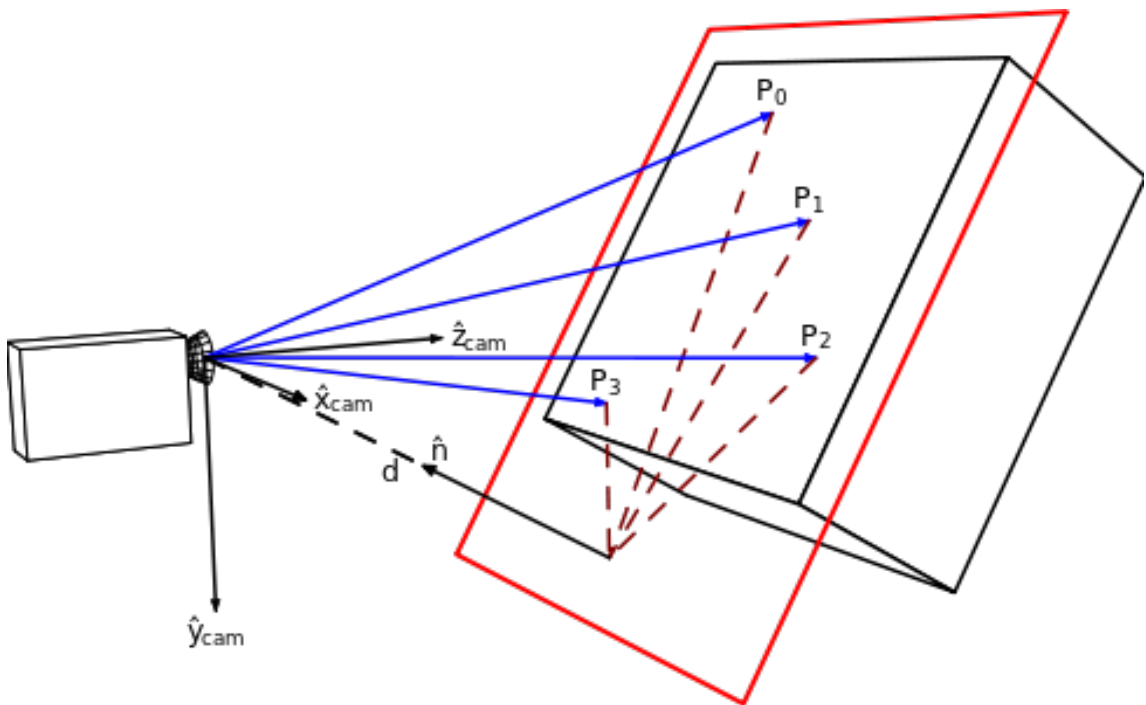


Fig.20 “Пример гомографии”

На рисунке 20 показана камера направленная на плоскость на расстоянии  $D$ . Мы можем вывести:

$$n^T P_i + d = 0$$

а качестве формулы плоскости и  $n^T P_i$  это проекция вектора  $P_i$  в  $n^T$ , и равная  $-d$ . Таким образом, мы можем написать

$$t = t\left(-\frac{n^T P_i}{d}\right)$$

Наконец, мы имеем

$$H_{ab} P_i = R P_i + t$$

где

$$H_{ab} = R - \frac{t n^T}{d}$$

Эта формула действительна только если камеры не имеет вращение и без перевода. В общем случае, когда  $R_a, R_b$  и  $t_a, t_b$  являются соответствующими поворотами и трансформациями камер  $a$  and  $b$ ,  $R = R_b R_a^T$  и матрица гомографии  $H_{ab}$  становится

$$H_{ab} = R_b R_a^T - R_b \frac{(t_a - t_b) n^T}{d} R_a^T = R_b \left( I - \frac{(t_a - t_b) n^T}{d} \right) R_a^T$$

где  $d$  является расстояние от камеры до плоскости. С помощью этой формулы можно оценить матрицу трансформации на основе точек на плоскости.

## Алгоритм

Чтобы решить эту проблему, я предложил алгоритм, состоящий из нескольких этапов. Алгоритм был реализован в Matlab для тестирования использование обнаружения ребер и частично в JavaScript для тестирования скорости обработки видео в реальном приложении. Псевдокод алгоритма приведен ниже.

Listing 1. Pseudo code of the algorithm proposed.

1. Image = binarizeImage(ImageRaw)
2. ImageTrace = traceImage(Image)
3. Edgelist = edge(ImageTrace)
4. Lines = edgeClustering(Edgelist)
5. Goodlines = union(Lines)
6. Points = intersect(Goodlines)
7. ImageTransformed = homography(Image, points)
8. ImageFinal = crop(Image\_transformed)
9. FeaturesBefore/After = detectSURFfeatures(ImageFinalBefore/After)
10. FeaturesMatched = matchFeatures(FeaturesBefore, FeaturesAfter)
11. FeaturesFilteredMatched = filterFeatures(FeaturesMatched)
12. ShiftField = extrapolateField(FeaturesFilteredMatched)
13. ImageFinalAfterShifted = shiftImage(ImageFinalAfter, ShiftField)
14. ImageSubtraction = subtractImages(ImageFinalAfterShifted, ImageFinalBefore)
15. ImageSubtractionBin = binarizeImage(ImageSubtraction)
16. SymbolContours = traceImage(ImageSubtractionBin)

Now I will describe each function presented in listing in more detail.

Fig.21 “UML-диаграмма”





## Бинаризация изображения

Бинаризованное изображение - это цифровое изображение, которое может иметь только два значения для каждого пикселя. Как правило, два цвета, используемые для бинарного изображения черно-белые, хотя любые два цвета могут быть использованы. Один цвет является цветом объекта переднего плана и остальная часть изображения является фоном.

Есть три основных алгоритма, который могут бинаризовать изображения: сегментация, порог и трансформация. Я решил использовать порог, потому что этот алгоритм обеспечивает очень высокую скорость и очень легко использовать.

Обычно базовая бинаризации не всегда обеспечивает соответствующие результаты из-за проблем с фоном. Основным недостатком здесь может быть то, что белый фон или темный документ может полностью испортить результаты. На рисунке видно, что бинаризация охватила часть фона.

Но, как вы увидите далее трассировка и пост обработка позволяют работать с неидеальными результатами бинаризации. Кроме трассировки и последующего обнаружения ребер может работать, даже если часть документа спрятана за руками пользователя.

Основной способ улучшить результаты этой части алгоритма состоит в использовании алгоритмы сегментации, которые могут помочь найти документ на изображении.

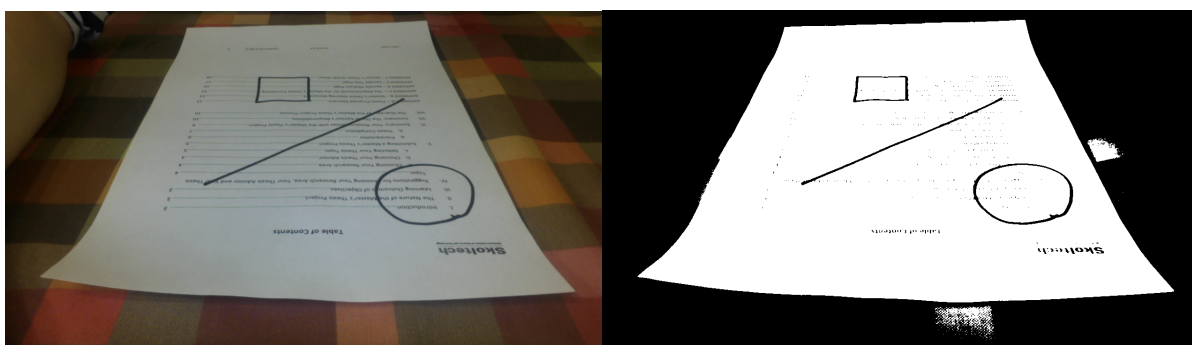


Fig.22 “Бинаризация изображения”

## Трассировка изображений

Векторизация - преобразование растровых изображений в векторные. Он может быть использован для создания изображения, пригодного для масштабирования без потери качества, а также, если дальнейшая обработка изображения будет выполняться на специфичном оборудовании (плоттеры, CNC).

Есть различные способы проведения векторизации. Я решил использовать метод трассировки, который работает следующим образом:

- 1) вы указываете начальное направление поиска для пикселей объекта, соседствующих с пикселем
- 2) затем алгоритм начинает идти от пикселя к пикселю, и когда он находит пиксель с ненулевым значением он начинает искать соседние пиксели с ненулевым значением.
- 3) если он находит пиксель, то он начинает искать соседа нового пикселя и так далее.
- 4) если он не найдет пиксель или пиксель уже посещен, то алгоритм рассматривает последовательность пикселей, как контур объекта, а затем продолжает со следующими пикселями.
- 5) в конце алгоритм возвращает объекты, которые определяются контурами.

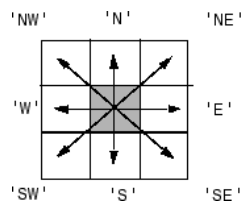


Fig.23 “Алгоритм трассировки”

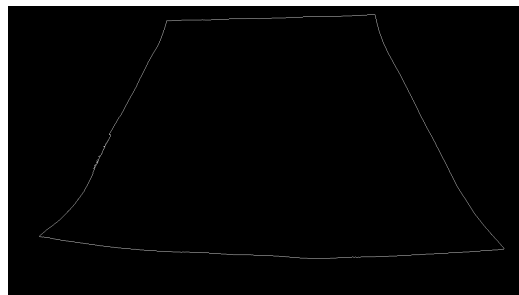


Fig.24 “Контур документа”

## Обнаружение ребер

Обнаружение ребер - набор математических методов, которые направлены на выявление точек в цифровом изображении, где яркость значительно изменяется или имеет разрывы.

Эти точки, где яркость меняется значительно организованы в набор отрезков.

Обнаружение ребер является очень важным инструментом в обработке изображений и компьютерном зрении, особенно в областях выявления признаков и выделения признаков.

Методы обнаружения края, как правило, организованы в две категории: на основе поиска и на основе пересечения нулевого уровня. Первый способ обнаруживает края путем расчета значения ребра, обычно производная первого порядка, например градиент величины, после чего находит направление максимума градиента с помощью вычисленной оценки ориентации ребра, например направление градиента. Методы нулевого пересечения находят нулевые пересечения с помощью производной второго порядка, рассчитанного по изображению, например с помощью метода Лапласа. Обычно некоторая предварительная обработка необходима для обнаружения ребер, например сглаживание, как правило, сглаживание Гаусса, почти всегда применяются. Я решил использовать алгоритм определения краев Кэнни, потому что он дает лучшие результаты, чем Собель, Превитт и Робертс (рис 25).

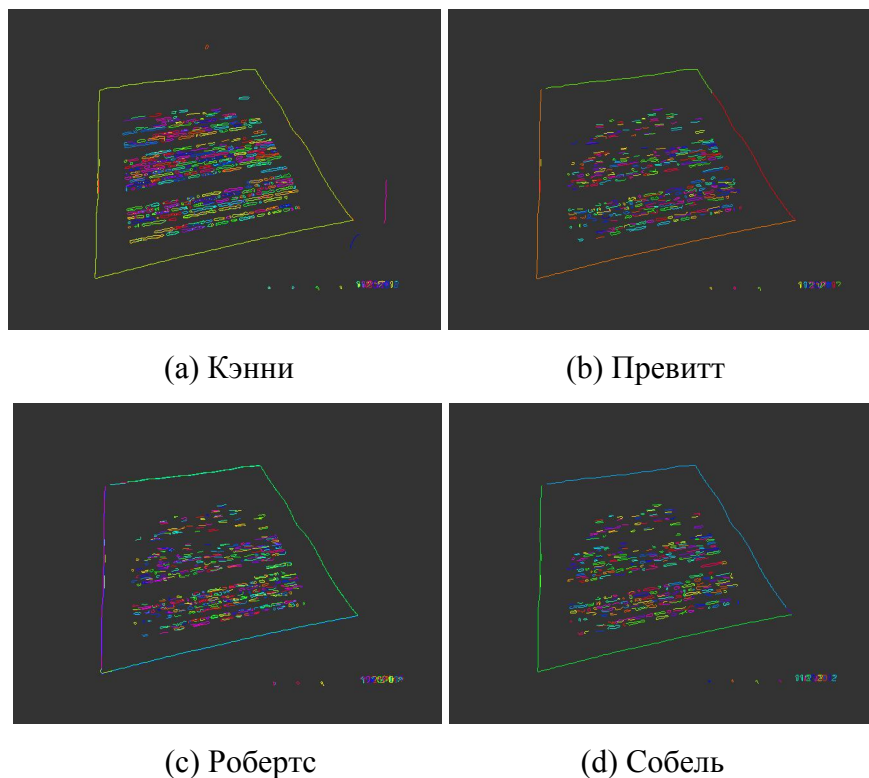


Fig. 25 Кэнни (a), Превитт (b), Робертс (c) и Собель (d) методы определения ребер.

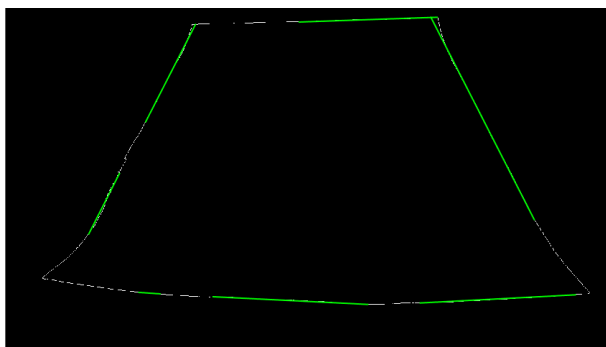


Fig.26 “Кэнни метод определения ребер”

## Кластеризация ребер

После первого шага, я получаю список всех ребер, обнаруженных в изображении. Этот список включает в себя потенциальных края куска документа, которые необходимо обнаружить. Эти края, как правило, являются самыми большими из списка, а также являются непрерывными, если рука или другие барьеры не разбивают их на подразделы. Я предполагал, что все найденные ребра связаны с краями документа. Чтобы понять, какое ребро связано с каким краем документом я использую кластеризации среди ребер.

Я решил использовать К-среднее, чтобы сделать это. Используя линии (линия =  $kx + b$ ) из списка ребер, которые я считаю претендентами на края документа. Тогда я использую  $(k,b)$  в качестве параметров каждого ребра и провожу кластеризацию среди них, используя эти параметры. Так как  $k$  и  $b$  имеют разные диапазоны, то есть необходимость нормализации этих параметров, которую я провожу на основе размера изображения.

К-среднее работает довольно быстро и надежно, что дает хорошие результаты, как вы можете видеть на рисунке.

Основные проблемы могут возникнуть, если мы имеем дело с препятствиями, но она может быть решена путем предварительной фильтрации и создания большего количества кластеров с последующей фильтрацией.

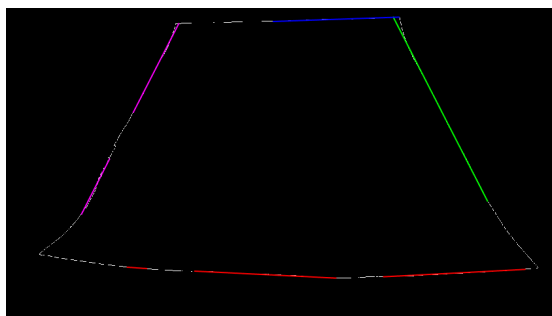


Fig.27 “Кластеризованные ребра”

## Объединение кандидатов

В этой функции, мне необходимо получить четыре линии, которые соответствуют краям документа. Для этого я комбинирую кластеризованные линии. Эта процедура необходима, потому что мы должны найти  $k$  и  $g$  параметры для всего края документа, имея эти параметры только с ребер. Я объединяю ребра, находя самые далекие точки для каждого кластера ребер. После этого я использую эти значения, чтобы получить край документа.

Несмотря на то, что есть некоторые ошибки, которые связаны с нелинейными сторонами документа на изображениях, мы можем использовать эту технику, потому что для гомографии мы не должны использовать точные углы документа, а углы, которые производятся крупнейшими краями, так как использование точных углов приводит к очень высокому коэффициенту ошибок.

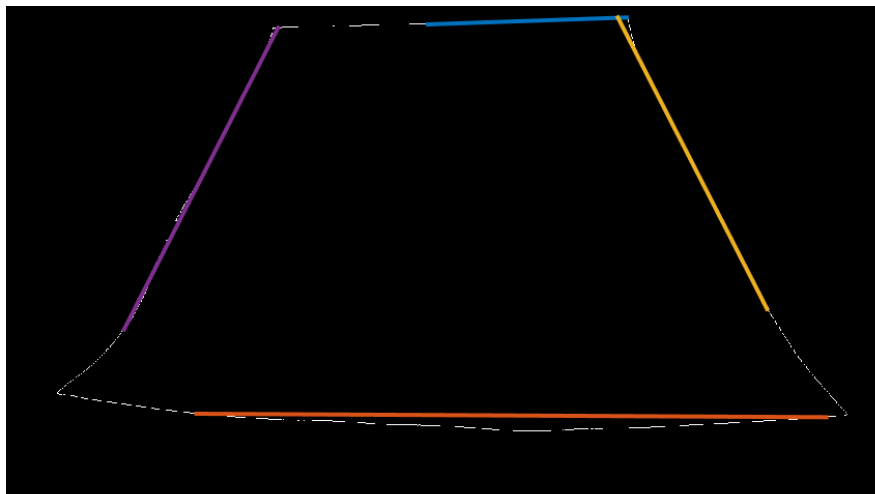


Fig.28 “Объединенные ребра”

## Получение точек пересечения линий

Эта функция получает точки пересечения линий рассматриваемых как краев документа. Использование этих точек я могу вычислить матрицу преобразования провести гомографии.

Как вы можете видеть на рисунке нижних точек не соответствуют очень хорошо corners документа. Причина этого в том, что для получения лучших результатов с гомографии мы

не делаем никакой необходимой необходимости использовать точные угловые координаты, но вместо того, что использование точек, которые производятся сторонами документа в качестве документа на изображения, как правило, загнали в угол стороны, когда они близки к угол.

Есть проблемы с такого рода методов, которые возникают, когда некоторые углы находятся за пределами изображения, а также случаев, когда огромная часть документа за какое-то препятствие.

Возможные улучшения включают в себя использование А4 Стандартные размеры, где мы можем оценить относительные размеры документа и положения углов с помощью стандартов размеров документа.

Наконец, мы можем использовать анализ серии изображений документа, чтобы отслеживать углах документа и, следовательно, предсказать положение там, даже если они скрыты за препятствием или лежащую в недоступном для камеры.

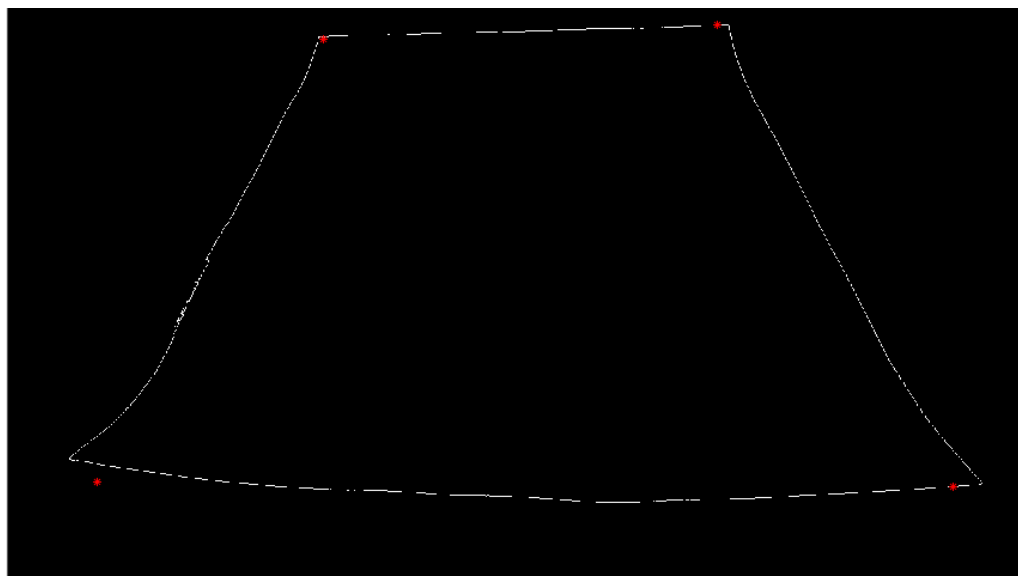


Fig.29 “Углы документа”

## Проведение гомографии

Гомография является центральной частью алгоритма, так как она производит изображение документа, которое очень похоже на отсканированное изображение. Хотя есть некоторые ошибки, которые возникают в связи с неточным определением сторон, результат, как вы можете видеть на рисунке очень хороший.

Все последующие шаги работают с изображением в градациях серого. Я решил использовать стандартный процесс перевода изображения в градации серого:

$$I = 0.2989 * R + 0.5870 * G + 0.1140 * B$$

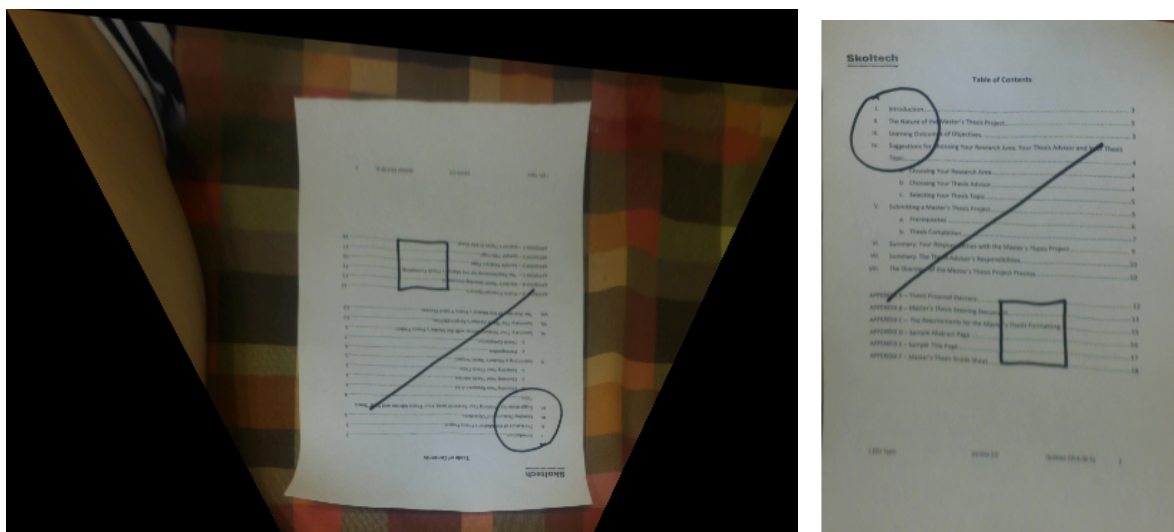


Fig.30 “Гомография”

## Обнаружение признаков

Идея обнаружения особенностей относится к способам, которые направлены на вычислении абстракции данных изображения и создания решений в каждой точке изображения, если есть функция изображения в этой точке или нет. Особенности результат в части изображения, как правило, они представлены как изолированных точек, непрерывных кривых или связанных областях.

Я SURF (ускорили надежными функциями) функции детектора, который местные детектор особенность. Это, в основном, вдохновленные масштабно-инвариантной функции преобразования (SIFT) дескрипторов. Реализации SURF быстрее, чем SIFT и объявил, чтобы быть более устойчивым к различным преобразованиям изображения, чем SIFT.

SURF использует целое приближение определителя Гессе детектора двоичных объектов, которые могут быть вычислены с тремя целочисленные операции с использованием целостный образ. SURF использует сумму вейвлет Хаара ответ вокруг точки интереса как особенностей. Они могут быть рассчитаны с помощью интегрального изображения.

SURF дескрипторов могут быть использованы для различных целей, таких как: найти и распознать объекты, узнавать людей или лица, создать 3D сцены, объекты следа и экстракт достопримечательности. SURF была создана Гербертом Беем.

Я считаю SURF детектор особенность, чтобы быть достаточно быстрым и надежным для этой задачи. Тем не менее некоторые пост-обработки требуется для алгоритма, поскольку он производит много шума точек.

Как вы можете видеть на картинке все функции, расположенной на части с текстом и элементов и проблемы, что, если вы начинаете выбирать функции, основанные только на их стоимости вы потеряете особенности, которые расположены рядом с границей документа. Я рассмотрел SURF детектор особенность, чтобы быть быстрым и достаточно прочный для этой задачи. Тем не менее некоторые пост-обработки требуется для алгоритма, поскольку он производит много шума точек.

Как вы можете видеть на картинке все функции, расположенной на части с текстом и элементов и проблемы, что, если вы начинаете выбирать функции, основанные только на их стоимости вы потеряете особенности, которые расположены вблизи границы документа..

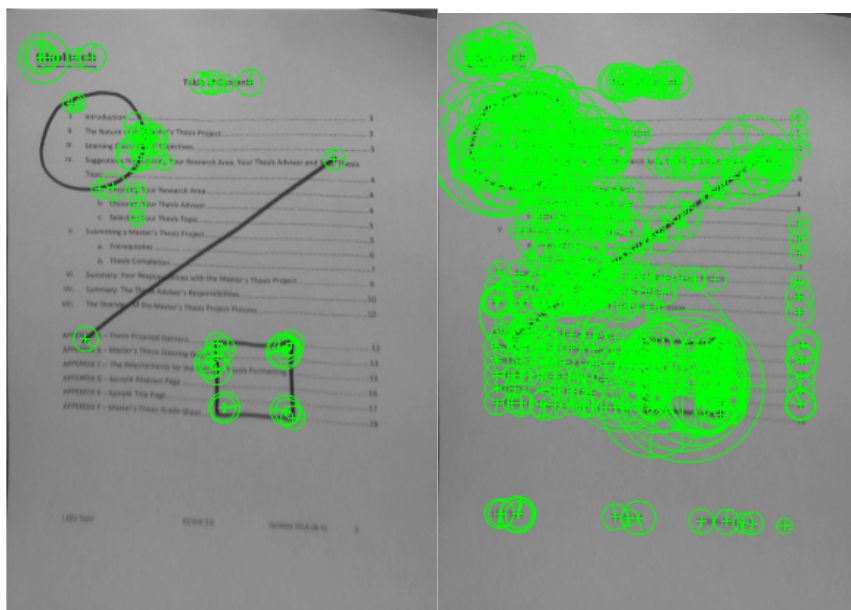


Fig.31 “SURF признаки”



## Соответствие признаков

Для векторных основе функций, таких как SIFT и SURF, существуют следующие методы сопоставления признаков:

### 1) Исчерпывающий

Этот метод вычисляет попарно расстояние между векторами признаков и использует его для математики им.

### 2) Приблизительный

Этот метод использует эффективный приближенное ближайший сосед поиска. Этот метод используется для больших наборов объектов. Я решил использовать метод исчерпывающего, так как это дает лучшие результаты, и не значительно увеличить время выполнения.

Существуют следующие параметры, которые используются для этого метода.

- 1) Порог, который представляет собой процент от расстояния от идеального матча. Два вектора признаков найдено, если расстояние между ними меньше, чем пороговое значение. Способ отклоняет матч, когда расстояние между функциями больше, чем значение порога. Увеличение этого значения приводит к большему количеству соответствующих признаков.
- 2) Коэффициент порога, который используется для фильтрации неоднозначных соответствующих признаков. Увеличение этого значения приводит к большему количеству соответствующих признаков.

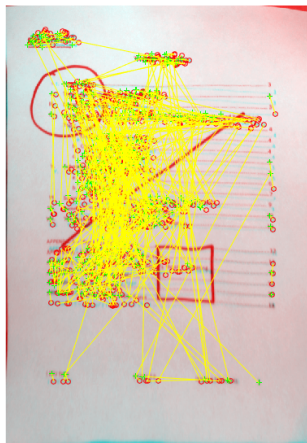


Fig.32 “Соответствие SURF признаков”

## Фильтрация признаков

Как мы можем видеть на изображении мы смогли найти много особенностей, которые позже был подобран. Используя все возможности приведет к большой погрешностью.

К особенностям фильтра я решил использовать два метода:

### 1) Фильтра по расстоянию

Прежде всего я убрал пары признаков, которые были слишком далеки друг от друга. Этот фильтр помогает удалить большинство пар и оставить только те, которые расположены ближе друг к другу.

### 2) Кластерный фильтр

Проблема с отфильтрованными признаками состояла в том, что много пар были близки друг к другу, но имели очень разные направления. Для фильтрации шумов, я использовал следующий алгоритм

- 1) для каждой точки найти точки в непосредственной близости
- 2) получить массив сдвига векторов точек
- 3) найти средний вектор среди массива векторов
- 4) сравнить средний вектор со сдвигом вектора исходной точке
- 5) если разница большая, то мы удаляем пару признаков

Эта фильтрация помогает получить только характерных точек, которые локально похожи, даже если не похожи на глобальном уровне.

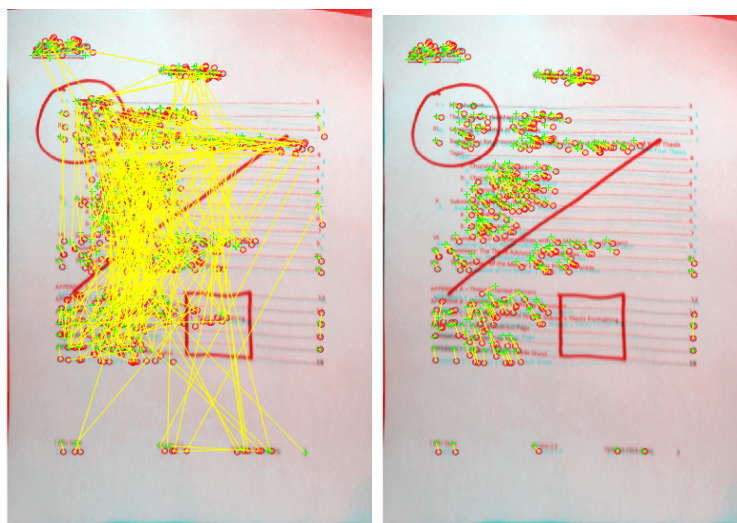


Fig.33 “Признаки до и после фильтрации.”

## Экстраполяция поля

Так как мы знаем только количество совпадающих точек на изображениях, мы должны найти способ, чтобы соответствовать все точки между изображениями. Для этого я использовал полевой подход. Мы можем представить все матчи в двух местах, один из которых будет иметь значение горизонтального сдвига для каждого пикселя, а другой будет иметь значение сдвига по вертикали для каждого пикселя.

Тогда задача нахождения матча для всех точек превращается в задачу экстраполяции и интерполяции поля на основе дискретного набора известных точек.

Существуют различные методы, которые могут быть использованы для экстраполяции и интерполяции.

- 1) Линейная интерполяция, где мы пытаемся найти градиент на границе корпуса, а затем оценить очки в зависимости от градиента.
- 2) Интерполяция по ближайшему соседу, где мы, чтобы найти значение в точке выбираем значение ближайшей точки и не используем значения соседних точек вообще.
- 3) Интерполяция по “натуральному” соседу, которая является методом пространственной интерполяции, разработанным Робинотом Сибсоном.

Я решил использовать линейную интерполяцию потому что она произвела лучшие результаты и работала довольно быстро.

Тем не менее, как вы можете видеть на рисунке экстраполяции, как правило, плохо работают на сторонах документа. Это не создает очень большую ошибку, так как там не много информации о сторонах документа, но если мы хотели бы улучшить этот метод, мы могли рассмотреть вопрос о включении нулевого сдвига на углах документа или на краях, которые были обнаружены в ходе предыдущих этапов.

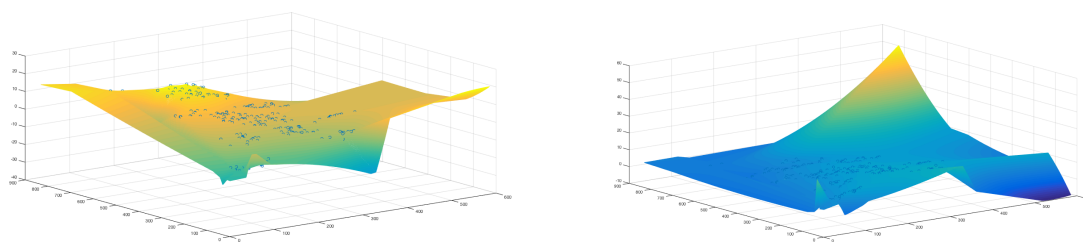


Fig.34 “Экстраполяция полей сдвига по вертикали и горизонтали”

## Трансформация изображения

Одним из основных частей предварительной обработки для вычитания преобразование изображения. Наиболее распространенные методы преобразования являются гомография и аффинное преобразование, которые используются, например, когда нужно объединить два изображения или построить панораму.

Проблема с преобразованием, что я не иметь дело с линейной трансформации здесь и из-за этого я представляю трансформацию как сдвиг поля, была оценена во время предыдущего блока алгоритма.

Как вы можете видеть, что есть огромное улучшение вычитается изображений с поля, приложенного сдвига.

Некоторые проблемы все еще возникают - как вы можете видеть на рисунке некоторые элементы изображения не вычесть полностью.

Основные способы повышения производительности является повышение предварительную обработку и добавление пост-обработки, которые будут обсуждаться ниже.

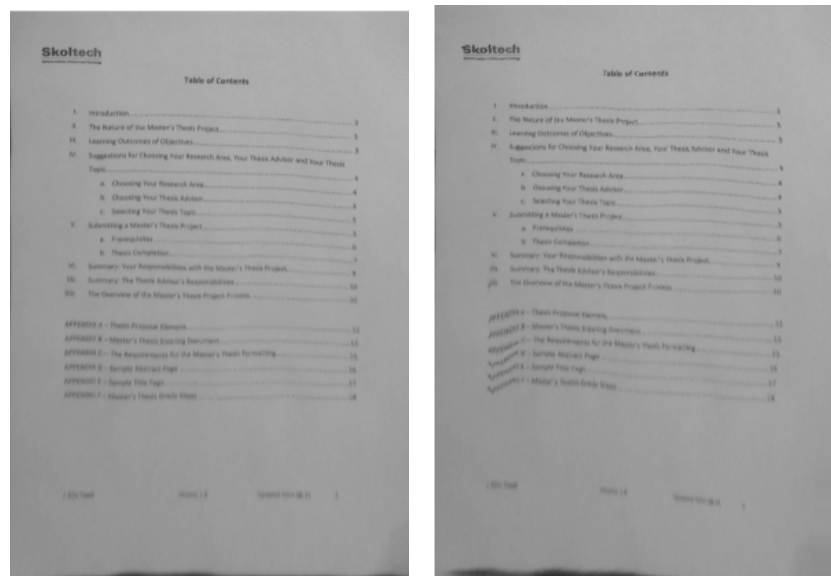


Fig.35 “Трансформация изображения”

## Вычитание изображений

Вычитание изображений - процесс, посредством которого цифровой числовое значение пикселей двух изображений вычитаются. Обычно это делается по одной из двух целей - выравнивание неровных участков (например, для удаления тени) или обнаружения изменений между двумя изображениями.

Основные проблемы, связанные с обнаружением изменений возникает, когда одни и те же элементы имеют различные позиции на изображениях до и после. Как вы можете видеть на рисунках есть огромная разница в вычитания изображений с и без смещения. В то время как некоторые ошибки по-прежнему на месте, большинство элементов вычитается, и мы можем себе ясного изменения между до и после настройки.

Есть способы, чтобы улучшить результаты вычитания. Основной способ сделать это дополнительная сканирование изображения и удаления точек с низким числом соседей с таким же значением. Дополнительная шаг может быть использование окна сканирования, который можно вычислить среднюю интенсивность пикселей в окне и четких окон с значений, близких к нулю, потому что, как вы можете видеть на своих ошибках изображения имеют противоположные значения и почти такой же площади. Именно поэтому окна сканирования может дать улучшения в алгоритм.

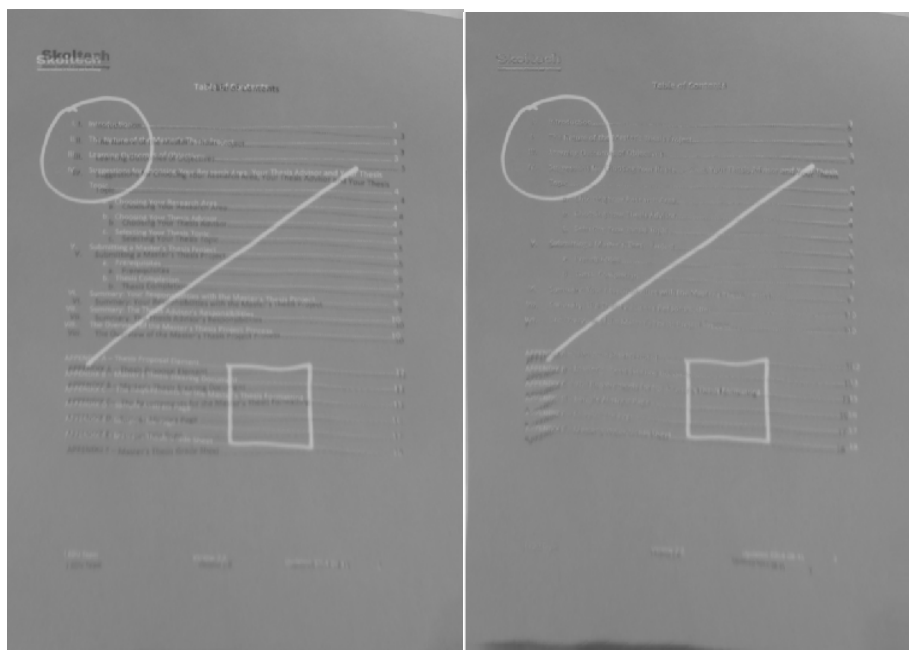


Fig.36 “Разница изображений без и с трансформацией”

## Бинаризация изображения

Я решил использовать метод порогового сравнения, поскольку он обеспечивает очень высокую скорость и очень легко осуществить.

Обычно бинаризация не будет хорошо работать для выявления изменений, где некоторые элементы изменили свою позицию. Основным недостатком здесь может быть, что некоторые фоновые неизменные элементы могут обеспечить шум.

Но потому что я заранее произвел сдвиг изображения я смог получить приличные результаты с пороговой бинаризации. Важно, чтобы получить очень хорошие результаты на этом этапе, потому что все не фоновые элементы будут применяться к цифровой версии документа и считаются знаки пользователя.

Основной способ улучшить результаты этой части алгоритма состоит в использовании пост-фильтрации на основе числа без фоновых соседей.

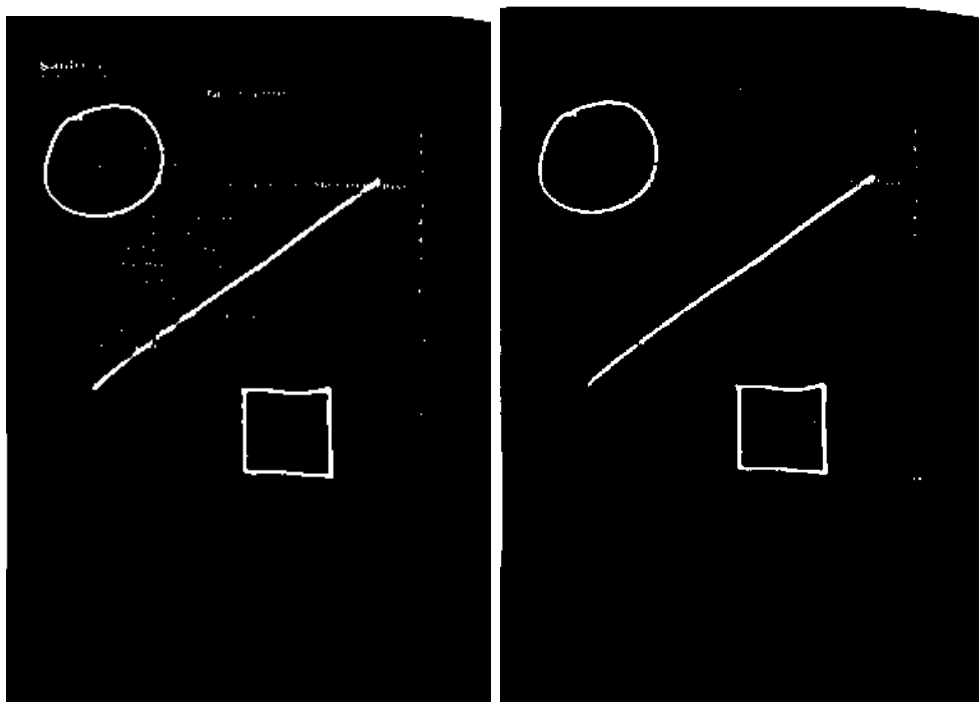


Fig.37 “Бинаризованная разница без и с трансформацией”

## Трассировка изображения

Векторизация - преобразование растровых изображений в векторные представления; обратный процесс растеризации. Он может быть использован для создания изображения, пригодного для масштабирования без потери качества, а также, если дальнейшая обработка изображения будет выполняться на конкретном оборудовании (плоттеры, CNC).

Есть различные способы проведения векторизации. Я решил использовать метод трассировки метод, который работает следующим образом:

- 1) вы указываете начальное направление поиска для пикселей объекта, соседствующих с пикселем
- 2) затем алгоритм начинает идти от пикселя к пикселю, и когда он находит пиксель с ненулевым значением он начинает искать соседние пиксели с ненулевым значением.
- 3) если он находит пиксель, то он начинает искать соседа нового пикселя и так далее.
- 4) если он не найдет пиксель или пиксель уже посещен, то алгоритм рассматривает последовательность пикселей, как контур объекта, а затем продолжает со следующими пикселями.
- 5) в конце алгоритм возвращает объекты, которые определяются контурами.

Skoltech  
SCHOOL OF MANAGEMENT

Table of Contents

I. Introduction.....	3
II. The Nature of the Master's Thesis Project.....	3
III. Learning Outcomes of Objectives.....	3
IV. Suggestions for Choosing Your Research Area, Your Thesis Advisor and Your Thesis Topic.....	4
a. Choosing Your Research Area.....	4
b. Choosing Your Thesis Advisor.....	4
c. Selecting Your Thesis Topic.....	5
V. Submitting a Master's Thesis Project.....	5
a. Prerequisites.....	6
b. Thesis Completion.....	7
VI. Summary: Your Responsibilities with the Master's Thesis Project.....	9
VII. Summary: The Thesis Advisor's Responsibilities.....	10
VIII. The Overview of the Master's Thesis Project Process.....	10
APPENDIX A - Thesis Proposal Element.....	12
APPENDIX B - Master's Thesis Steering Document.....	13
APPENDIX C - The Requirements for the Master's Thesis Formatting.....	15
APPENDIX D - Sample Abstract Page.....	16
APPENDIX E - Sample Title Page.....	17
APPENDIX F - Master's Thesis Grade Sheet.....	18

| EDU Team                      Version 2.0                      Updated 2014-08-31                      2

Fig.38 “Цифровой документ с цифровыми пометками”

# Реализация с помощью Javascript

## Мотивация

Поскольку этот проект направлен на использование в реальном приложении я решил провести серию испытаний на пользователях, чтобы узнать больше о потребностях пользователей и их проблемах. Основными направлениями использования этого приложения являются:

- 1) Дистанционное образование
- 2) Видеолекции
- 3) Проектор

Я решил выбрать дистанционное обучение для проверки пользователей, потому что:

- 1) рынок наполнен мелкими игроками, которые открыты для новых технологических инноваций
- 2) пользователи молоды и имеют более высокую скорость изучения новых технологий

Чтобы проверить проект с реальными пользователями, я решил сделать прототип с помощью Javascript. Это позволило мне не только создать очень масштабируемый продукт, который легко развернуть, но также сделать его знакомым для многих пользователей.

Я решил проанализировать состояние рынков, и увидел мне, что основные проблемы для этого вида программного обеспечения были:

- 1) Разрешение веб-камеры, которое очень важно, так как только 640x480 и лучше способны производить хорошие результаты.
- 2) Ограничения браузера
- 3) Ограничения CPU, которые важны, так как программа должна проводить обработку изображения с высоким разрешением для получения результата высокого качества.

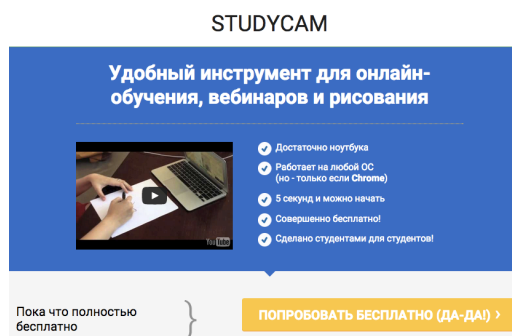


Fig.39 “Веб-сервис – главная страница”



## Алгоритм

Реализованное демо можно найти на [studycam.ru](http://studycam.ru)

Есть следующие требования к использованию демо:

- 1) Chrome / Opera / Яндекс браузеры, которые необходимы, так как они являются единственными, которые могут получить доступ к веб-камере с помощью Javascript
- 2) Установленный Adobe Flash Player, который обеспечивает необходимые библиотеки для браузеров для доступа к веб-камере

Я использовал следующие библиотеки для реализации обработки в реальном времени:

- 1) Jsfeat (<https://inspirit.github.io/jsfeat/>), которая является главной библиотекой и имеет ряд полезных методов обработки изображений и позволяет работать с матрицами и потоком видео.
- 2) Pixastic (<https://github.com/jseidelin/pixastic/commits/master>), которая имеет много полезных функций обработки изображения для удаления шума, обнаружения ребер и так далее.

Я организовал интерфейс следующим образом:

- 1) Целевая страница, где пользователь может видеть следующие элементы
  - a. Видео, которое описывает, как продукт работает и как нужно наклонить стенку ноутбука, чтобы использовать его
  - b. Описание и требования, которые описывают, какой браузер Вы должны использовать и другие параметры
- 2) Страница с видео
  - a. Тестирующая страница, которая анализирует тип браузера, и если он не удовлетворяет требованиям, то она дает ссылки на браузеры, которые удовлетворяют требованиям.
  - b. Предварительная загрузка страницы, которая показывает, какие действия нужно должны сделать, чтобы запустить программу, давая доступ браузеру.
  - c. Страница с видеопотоком, которая просит вас поставить документ в нужном положении, чтобы результат был хорошим.
  - d. Страница с преобразованным видеопотоком, которая показывает вам документ и позволяет изменять его масштаб с помощью клавиатуры.

Приведенный прототип был в состоянии работать достаточно быстро, чтобы процесс обработки видеопотока работал без каких-либо задержек, что было отмечено многими пользователями.

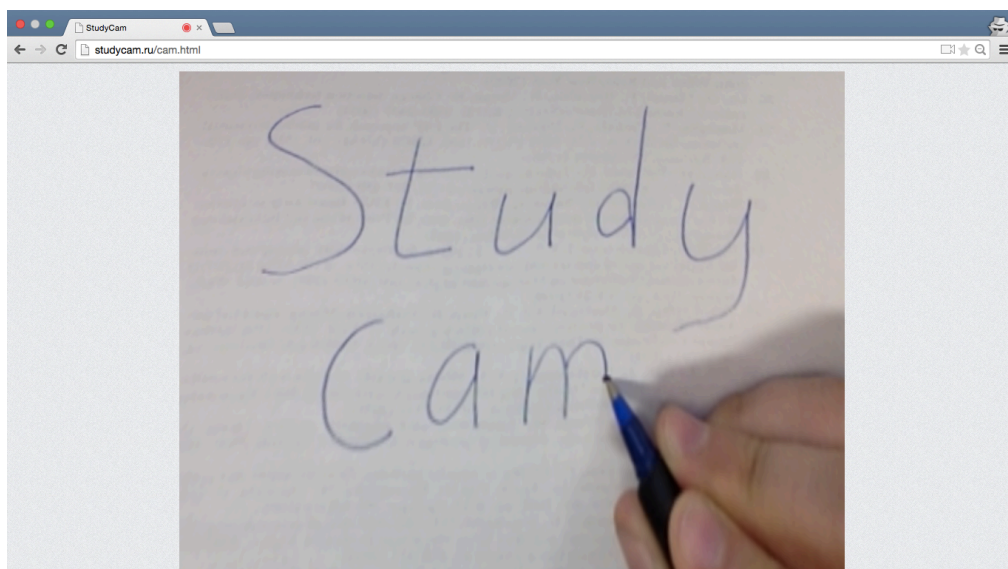


Fig.40 “Веб-сервис – страница с камерой”

## Пользовательское тестирование

Я провел тестирование с тремя группами онлайн обучения. Группы имели разные направления - математика, психология, рисование.

Основная обратная связь этих групп была очень позитивной. Прототип был в состоянии решить большую часть их проблем.

Я провел ряд интервью и я выяснил следующие проблемы:

- 1) Разрешение веб-камеры, потому что не все из них имеют современные камеры. Эта проблема очень фундаментальная и ее трудно решить
- 2) Дизайн пользовательского интерфейса, который был непонятен для некоторых из них. Я начал изменять интерфейс и использовал Яндекс метрику, чтобы лучше понять взаимодействие пользователей с интерфейсом
- 3) интеграция с Skype, потому что большинство из них используют эту программу, и они хотели бы, продукт, который интегрирован в него. Эта проблема может быть решена только путем переписывания программы, как плагин для скайп или приложение для ноутбука, которое будет эмулировать веб-камеру
- 4) ограничения браузера, потому что не все используют требуемый браузер. Они загрузили необходимый браузер и проблема была решена
- 5) интеграция с доской. Эта проблема является скорее предложением о добавлении функциональности совместной доски, где вы можете рисовать с помощью мыши и ваш собеседник будет видеть результаты

Главным достижением тестирования с моей точки зрения является то, что одна из групп решила включить веб-сервис в процессе их образования.

## Результаты

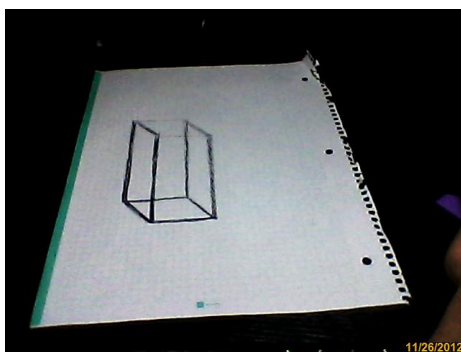
В этом разделе представлены результаты. Алгоритм работает отлично, и результат его работы показан на рисунке.

Но есть случаи, на которых алгоритм не работает. Некоторые примеры показаны на рисунке. Для фигуры (a) причиной является неровные края (линия, которая представляет потенциальный край не может быть построена). На рисунке (b) причиной плохой работы является то, что почти весь документ скрыт за руками, и, следовательно, нет достаточно информации, чтобы сформировать края.

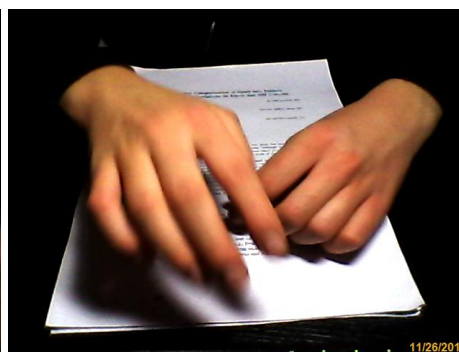
Таким образом я могу перечислить следующие проблемы, которые моя текущая реализация не учитывает:

- 1) препятствия, которые могут быть решены с помощью серий изображений с веб-камеры для отслеживания стороны документа
- 2) неровные края, которые могут быть решены путем применения дополнительных фильтров к изображению, чтобы удалить эффект неровности
- 3) белый фон, который может быть решен с помощью контрастной нормализации, которая позволит выделить фон и документ

В конце концов стоит отметить, что многое может быть сделано для улучшения качества реализации, но в конечном счете текущая реализация прекрасно работает с большинством случаев. Пользовательское тестирование показало, что основные проблемы для пользователей лежат не в области алгоритма, а в области пользовательского интерфейса и пользовательского опыта. Я буду продолжать работать в этом направлении, чтобы улучшить качество продукта для пользователей.



(a) неровный край



(b) спрятанный край

Fig.41 “Пример, когда алгоритм не работает”.



(a) вид пользователя с приложением (b) изображение полученное с камеры

Fig. 42 “Javascript реализация”

## Заключение

В этом проекте, я нашел проблему, которая возникает в электронной связи, когда люди должны использовать бумажный носитель, чтобы объяснить или показать что-то для своего собеседника в разговоре, а также когда требуется отсканировать документ имея только ноутбук. На данный момент нет хороших способов решить эту проблему. Одной из причин этого является то, что разрешение камеры в ноутбуке не всегда позволяло производить результаты с высоким качеством.

Чтобы решить проблему, я предложил метод, который позволяет конвертировать и передавать видеопоток таким образом, что приемник может увидеть документ, как если бы он сидел рядом с отправителем и сканировать изменения на документ, как если бы у вас был сканер. Я описал предлагаемый алгоритм, реализовали его на двух языках - Matlab и Javascript, проанализировал результаты и показал случаи, в которых мой алгоритм работает хорошо и случаи, в которых она функционирует плохо. Для тяжелых случаев я предложил некоторые пути повышения производительности алгоритма.

Также я провел тестирование с реальными пользователями, которые помогли мне выявить потребности и проблемы. Эта информация имеет решающее значение для создания правильного продукта для рынка. Я планирую добавить дополнительные функции в продукт и сделать его открытым.

Как конечный результат, я могу сказать, что поле использования веб-камеры для сканирования документов неисследованно до конца и содержит много интересных идей и предложений.

## ССЫЛКИ

- [1] Liu, Y., Ikenaga, T., Goto, S., “An MRF model-based approach to the detection of rectangular shape objects in color images”, <http://www.hcii-lab.net/gpen/download/An%20MRF%20model-based%20approach%20to%20the%20detection%20of%20rectangular.pdf>
- [2] Micusik, B., Wildenauer, H., Kosecka, J., “Detection and Matching of Rectilinear Structures”, [http://www.researchgate.net/publication/224323260\\_Detection\\_and\\_matching\\_of\\_rectilinear\\_structures/file/5046351c827054fdf4.pdf](http://www.researchgate.net/publication/224323260_Detection_and_matching_of_rectilinear_structures/file/5046351c827054fdf4.pdf)
- [3] Shaw, D., Barnes, N., “Perspective Rectangle Detection”, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.7972&rep=rep1&type=pdf?hash=2n3rp89wf>
- [4] Noronha, S., Nevatia, R., “Detection and Modeling of Buildings from Multiple Aerial Images”, <http://iris.usc.edu/outlines/papers/2001/san-pami-01.pdf>
- [5] Jung, C., Schramm, R., “Rectangle Detection based on a Windowed Hough Transform”, <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.59.4239&rep=rep1&type=pdf?hash=asasidf98823hhrhp98ep>
- [6] Herley, C., “Recursive Method to Detect and Segment Multiple Rectangular Objects in Scanned Images”, <https://research.microsoft.com/pubs/70037/tr-2004-01.pdf>
- [7] Nishida, K., Fujiki, J., Tsushiya, C., Tanaka, S., Kurita, T., “Road plane detection using differential homography estimated by pair feature matching of local regions”, <http://home.hiroshima-u.ac.jp/tkurita/papers/nishida-SPPRA2011S.pdf>
- [8] Paul L. Rosin, Thresholding for Change Detection, <https://users.cs.cf.ac.uk/Paul.Rosin/resources/papers/thresh-ICCV.pdf>
- [9] Pierre-Luc St-Charles, Guillaume-Alexandre Bilodea, Robert Bergevin, A Self-Adjusting Approach to Change Detection Based on Background Word Consensus, <http://www.polymtl.ca/litiv/doc/StCharlesetalWACV2015.pdf>
- [10] R. B. Fisher, Change Detection in Color Images, <http://homepages.inf.ed.ac.uk/rbf/PAPERS/iccv99.pdf>

- [11] Lorenzo Bruzzone, Diego Fernández Prieto, Automatic Analysis of the Difference Image for Unsupervised Change Detection, [http://perso.telecom-paristech.fr/~angelini/SI241/papers\\_for\\_project/R17-CD-EM-Gauss.pdf](http://perso.telecom-paristech.fr/~angelini/SI241/papers_for_project/R17-CD-EM-Gauss.pdf)
- [12] Chiung-Yao Fang, Sei-Wang Chen, Automatic Change Detection of Driving Environments in a Vision-Based Driver Assistance System, <http://www.csie.ntu.edu.tw/~fuh/personal/AutomaticChangeDetectionofDrivingEnvironments.pdf>
- [13] Marcus Liwicki, Horst Bunke, Handwriting Recognition of Whiteboard Notes, <http://www.dfki.uni-kl.de/~liwicki/pdf/LiBu05-01.pdf>
- [14] Alex Graves, Jurgen Schmidhuber, Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks, <http://papers.nips.cc/paper/3449-offline-handwriting-recognition-with-multidimensional-recurrent-neural-networks.pdf?hash=89fhp23rpn98h2fpa>
- [15] Rahul Kala, Harsh Vazirani, Anupam Shukla, Ritu Tiwari, Offline Handwriting Recognition using Genetic Algorithm, <http://ijcsi.org/papers/7-2-1-16-25.pdf?hash=p28rfnp3f9pn398fn8>
- [16] Alex Graves, Santiago Fernandez, Marcus Liwicki, Horst Bunke, Jurgen Schmidhuber, Unconstrained Online Handwriting Recognition with Recurrent Neural Networks, [http://people.idsia.ch/~juergen/nips\\_2008.pdf?hash=98hwfo938hrpfnwe9u8fhp948f3p98fasd](http://people.idsia.ch/~juergen/nips_2008.pdf?hash=98hwfo938hrpfnwe9u8fhp948f3p98fasd)
- [17] Claus Bahlmann, Bernard Haasdonk, Hans Burkhardt, On-line Handwriting Recognition with Support Vector Machines-A Kernel Approach, [ftp://ftp.informatik.uni-freiburg.de/papers/lmb/ba\\_ha\\_bu\\_iwfhr02.pdf](ftp://ftp.informatik.uni-freiburg.de/papers/lmb/ba_ha_bu_iwfhr02.pdf)
- [18] Richard O. Duda, Peter E. Hart, Use of the Hough Transformation to Detect Lines and Curves in Pictures, <http://www.ai.sri.com/pubs/files/tn036-duda71.pdf?hash=jp198enp23np2>
- [19] Parmeshwar Khurd, Leo Grady, Rafiou Oketokoun, Hari Sundar, Tejas Gajera, Summer Gibbs-Strauss, John V Frangioni, Ali Kamen, Global error minimization in image mosaicing using graph connectivity and its applications in microscopy, <http://www.jpathinformatics.org/article.asp?issn=2153->

3539;year=2011;volume=2;issue=2;spage=8;epage=8;aualast=Khurd;type=3?id=test1ugbe  
b387bo3

[20] Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis  
and Machine Intelligence, 8(6):679–698, 1986

[21] Thomas Baltzer Moeslund, Canny edge detection,  
<http://www.cse.iitd.ernet.in/~pkalra/csl783/canny.pdf>