Федеральное агентство по образованию

**ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

Факультет - Приборостроительный

Кафедра - Автоматика и управление

ПРОЕКТ ПРОВЕРЕН ДОПУСТИТЬ К ЗАЩИТЕ

Рецензент Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

“\_\_\_\_\_” \_\_\_\_\_\_\_\_\_\_\_\_\_ 2007\_\_ г. “ \_\_\_\_\_ “ \_\_\_\_\_\_\_\_\_\_\_\_ 2007\_\_ г.

Программное обеспечение системы обработки изображения в реальном времени

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ДИПЛОМНОМУ ПРОЕКТУ**

**ЮУрГУ-Д.220200 068.000.ПЗ**

Консультанты: Руководитель проекта:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

“ \_\_\_\_\_ “ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2007 \_ г. Автор проекта

студент группы ПС-269м .

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пушников А.А. .

“ \_\_\_\_ “ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2007 \_ г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Нормоконтролер

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

“ \_\_\_\_ “ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2007 \_ г. “ \_\_\_\_\_ “ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2007 г.

Челябинск

2007 г.

# ВВЕДЕНИЕ

**АНАЛИЗ ЗАДАЧИ. ОБЗОР СУЩЕСТВУЮЩИХ МЕТОДИК РЕШЕНИЯ**

**1. ВЫДЕЛЕНИЕ ОБЪЕКТА НА ИЗОБРАЖЕНИИ**

# 1.1. ОБЩИЙ АЛГОРИТМ СРАВНЕНИЯ ДВУХ ИЗОБРАЖЕНИЙ

1.2. МЕТОД МАКСИМАЛЬНЫХ ПЛОЩАДЕЙ

1.3. МЕТОД ГИСТОГРАММ

1.4. ПОДГОТОВКА ИЗОБРАЖЕНИЯ К РАСПОЗНАВАНИЮ

**2. ЗАДАЧА РАСПОЗНАВАНИЕ ОБЪЕКТА**

2.1. МЕТОД НАИМЕНЬШИХ КВАДРАТОВ

2.2. МОДЕЛИРОВАНИЕ МНОГОМЕРНОЙ ФУНКЦИИ. РАСПРЕДЕЛЕНИЯ ВЕКТОРОВ, ПРИЗНАКОВ ИЗОБРАЖЕНИЯ ОБЪЕКТА

*Факторный анализ (FA)*

*Метод главных компонент (PCA)*

*Анализ независимых компонент (ICA)*

*Линейный Дискриминантный анализ (Linear Discriminant Analysis, LDA)*

2.3. ДЕФОРМИРУЕМЫЕ МОДЕЛИ.

2.4. СКРЫТЫЕ МАРКОВСКИЕ МОДЕЛИ (Hidden Markov Models, HMM)

2.5. МЕТОД ОПОРНЫХ ВЕКТОРОВ (Support Vector Machines, SVM)

**3. РЕАЛИЗАЦИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**ЗАКЛЮЧЕНИЕ**

# Введение

В настоящее время вычислительная техника используется во многих областях человеческой деятельности, являясь удобным и многофункциональным инструментом решения широкого круга задач. Многие отрасли техники, имеющие отношение к получению, обработке, хранению и передаче информации, в значительной степени ориентируются в настоящее время на развитие систем, в которых информация имеет характер изображений. Изображение, которое можно рассматривать как двумерный сигнал, является значительно более емким носителем информации, чем обычный одномерный (временной) сигнал. Вместе с тем, решение научных и инженерных задач при работе с визуальными данными требует особых усилий, опирающихся на знание специфических методов, поскольку традиционная идеология одномерных сигналов и систем мало пригодна в этих случаях. В особой мере это проявляется при создании новых типов информационных систем, решающих такие проблемы, которые до сих пор в науке и технике не решались, и которые решаются сейчас благодаря использованию информации визуального характера.

Цель машинного зрения - принятие решений о реальных физических объектах и сценах, основываясь на воспринимаемых изображениях. Машинное зрение теснейшим образом взаимодействует с областью обработки изображений, часто трудно однозначно отнести возникающие задачи и применяемые методы решения к одной из этих областей.

# Анализ задачи. Обзор существующих методик решения

Постановка задачи:

Разработать методы поиска на изображении, получаемом с видеокамеры, детали, с дальнейшей идентификацией ее к одной из трех групп, с целью автоматизации технического процесса.

Задачу можно разделить на три основных этапа:

1. Выделение объекта на изображении.
2. Подготовка изображения к распознаванию.
3. Задача распознавание объекта.

# Для решения каждой подзадачи, в настоящее время, существует множество различных методик.

# 1. Выделение объекта на изображении

Для успешного решения задачи распознавания, необходимо выделить искомый объект на изображении, и привести его к нормализованному виду, пригодному для распознавания.

Основной метод нахождения объекта на изображении – сравнение исследуемого изображения с эталонным. При конечной его реализации, возможны значительные изменения в алгоритме нахождения объектов.

Критерием успешности выполнения алгоритма, могут служить:

1. Оценка успешности человеком.
2. Анализ количества найденных объектов.
3. Анализ размеров найденных объектов.
4. Анализ расположения найденных объектов.
5. Анализ цветов найденных объектов.

**1.1.** **Общий алгоритм сравнения двух изображений**

Определим эталонное изображение, как множество точек , а исследуемое изображение, как множество . Пусть - функция яркости точки изображений, определённая на множествах и . Одним из распространённых подходов для определения областей движения между двумя кадрами изображения и основывается на сравнении соответствующих точек этих двух кадров. [3],[4]. Для этого применяется процедура формирования «разности» кадров. Разностью между двумя изображениями, является следующее множество:



где - значение порогового уровня, выбираемое таким образом, чтобы отделить точки, в которых кадры изображения значительно отличаются друг от друга, как правило это искомые объекты, от точек, в которых яркость изменилась незначительно, в связи с мерцание освещения, вибрацией камеры и другими помехами. Изображение содержит предположительные области движения объектов и аддитивный шум, искажающий текущий кадр. Избавится от шума, позволяют морфологические операции (операции над бинарным изображением), такие как эрозия, коррозия, а так же различные способы фильтрации и оптимальное (экспериментально подбираемое) значение порога .



Далее, можно обработать найденные объекты, для этого используются:

1. Метод максимальных площадей.
2. Метод гистограмм.

Алгоритм был реализован с помощью библиотеки для обработки изображений OpenСV. При тестировании алгоритма использовались изображения игровой доски для игры в го с белыми и чёрными камнями. Пример его работы представлен на рис .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Эталонное изображение | Исследуемое изображение | Разностное изображение | Бинаризированое изображение , при =10 | Отфильтрованное изображение |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |

Рис. Пример работы алгоритма.

## В первом столбце находятся изображения доски, являющиеся эталонными. Во втором столбце – исследуемыми. Из третьего столбца видно как изменилось положение камней на доске. В четвёртом и пятом столбцах показано соответственно результат бинаризации изображения третьего столбца и результат последовательных применений операций коррозии и эрозии к изображению в четвёртом столбце.

## 1.2. Метод максимальных площадей

Метод приводит найденные объекты к форме, более удобной для дальнейшей обработки, и объединяет отдельные области на разностном изображении в области по признаку их близости друг другу [].

Необходимо выделить области правильной формы, являющиеся достаточно крупными объектами. Задачу можно формализовать следующим образом: необходимо разбить множество точек на максимально возможное число непересекающихся подмножеств .



Предложим следующий алгоритм решения:

1. На множестве конструируем подмножества , содержащие связанные точки, выбранные как лежащие рядом друг с другом.



2. Конструируем новые подмножества , охватывающие те подмножества , которые лежат рядом друг с другом.



3. Повторяем пункт 2 до тех пор, пока не получим конечное количество подмножеств .



Метод был адаптирован и реализован функциями библиотеки OpenCV. Пример его работы представлен на рис.

|  |  |  |  |
| --- | --- | --- | --- |
| № | Разностное изображение | Результат: области правильной формы | |
| Круг | Прямоугольник |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |

Рис. Пример работы алгоритма для разных областей правильной формы.

В первом столбце показаны изображения, полученные после алгоритма сравнения двух изображений. Во втором и третьем столбцах показаны результаты работы вышеописанного алгоритма. Во втором столбце в результате преобразований мы получаем область в виде круга, а в третьем в виде прямоугольника.

В первой и второй строке исходным является изображение руки. Как видно, область движения руки не является однородной. В результате работы алгоритма, область движения руки сводиться к однородной области правильной формы (круг, прямоугольник).

## В третьей строке исходным является изображение камня на игровой доске. Аналогично, в результате работы алгоритма область движения камня сводиться к однородной области правильной формы (круг, прямоугольник).

## 1.3 Метод гистограмм

В методе используется гистограмма изображения искомого объекта для нахождения объекта с такими же цветовыми характеристиками на серии изображений.

Нужно построить изображение в оттенках серого цвета, содержащее необходимые нам объекты.

Введём оператор, который преобразует функцию яркости изображения в функцию количественного распределения пикселей с определенным значением яркости (гистограмму) (где k – численное значение яркости):



Обратный оператор преобразует гистограмму в изображение в оттенках серого.



Алгоритм состоит из следующих этапов:

1. Построение гистограмм искомого объекта и исходного изображения



1. Формируем новую гистограмму, как нормированное произведение и :



1. Используя обратное преобразование , получаем двумерную функцию, которая является искомым изображением в оттенках серого:



Метод был адаптирован и реализован функциями библиотеки OpenCV. Результаты применения метода приведены на рис.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | Доска | Черный камень | Белый камень |
| Искомое изображение | |  |  |  |
| Гистограмма искомого изображения | |  |  |  |
| Исходное изображение | | По гистограмме доски | По гистограмме черного камня | По гистограмме белого камня |
| № |  |  |  |  |
| 1 |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

Искомыми изображениями являются изображения игровой доски, чёрного и белого камня. В таблице представлены их гистограммы.

## Во всех трёх опытах к исходному изображению, содержащему область движения, применялся вышеописанный метод. В результате в каждом из опытов были получены три изображения. Каждое из изображений содержит область, в которой нахождение искомого объекта максимально, т.е. максимально количество белых пикселей в этой области

## 1.4. Подготовка изображения к распознаванию

С точки зрения задачи распознавания, более удобно использовать изображения объектов, имеющие одинаковый размер и приблизительно одинаковую ориентацию в пространстве. Однако, алгоритмы выделения объектов, возвращают объекты, искаженные перспективой – различных размеров и произвольно ориентированных на изображении.

Для приведения изображения найденного объекта к общему виду, необходимо повернуть его на нужный угол. В эталонных и исследуемых изображениях объектов находятся две контрольные точки, после чего изображения разворачивают, так чтобы вектора, соединяющие эти точки, совпали.

Контрольными точками могут быть, например:

1. Видимый центр изображения.
2. Центр масс изображения.
3. Точка, заметно отличающаяся от остальных по цвету.
4. Центр маркера, поставленного на объекте

и др.

Также, необходимо, привести эталонные и исследуемые изображения к одному размеру.

Перечисленные выше операции выполняются аффинными преобразованиями над матрицами изображений, общий вид которых:



Используются частные случаи аффинных преобразованияй:

1. Растяжение (сжатие) вдоль координатных осей, задаваемое в виде:



Растяжению вдоль соответствующей оси соответствует значение масштабного множителя большего единицы. В однородных координатах матрица растяжения (сжатия) имеет вид

.



1. Поворот вокруг начальной точки на угол , описываемый формулой:



Матрица вращения (для однородных координат)

.



1. Перенос, задаваемый простейшими соотношениями:



Матрица переноса имеет вид

.



# 2. Задача распознавание объекта

Данное раздел включает в себя широкий класс задач, различающихся в основном тем, каким образом задаются характеристики объекта и в как требуется его классифицировать.

Методы, применяемые для решения поставленной задачи, во многом зависят от особенностей объекта, который требуется локализовать. Зачастую, постановка задачи неформальна – описать свойства нужного объекта в математических терминах бывает достаточно сложно, поэтому задание часто звучит, например, так - нужно найти на изображении все, похожее вот на "это" (картинка с примером). Или, даже просто словами – найти на изображении всех, скажем, божьих коровок. Соответственно, решение задачи заключается в формулировке свойств распознаваемого объекта и конструировании устойчивого метода нахождения объектов, отвечающих указанным свойствам.

В числе основных сложностей при решении данной задачи – большое разнообразие входных данных и трудность выделения общих свойств внешнего вида для объектов естественного происхождения. Объекты искусственного происхождения обычно распознавать значительно легче.

В методах описания свойств объекта для нахождения можно выделить два крайних направления:

***Обобщение и использование эмпирических данных и правил об объекте (top-down, bottom-up)***

Идея заключается в нахождении, обобщении и формулировке в математических терминах эмпирических наблюдений и правил о том, как на изображениях обычно выглядит интересующий нас объект. Продолжая пример с божьей коровкой, можно подметить следующее:

1. Божьи коровки обычно рыжего или красного цвета;
2. На спине у них обычно присутствует некоторое количество черных пятнен (можно также посчитать примерное соотношения размера пятен с размером насекомого);
3. Спина у них разделена на две половинки темной линией, обычно видимой. С одной из сторон этой линии у божьей коровки голова – темная, соотносящаяся по размерам с телом в некоторой пропорции;
4. Сверху божья коровка выглядит примерно как эллипс;

Хорошо, если известны дополнительные условия задачи и получения входных изображений, например:

1. Приблизительно известны ожидаемые размеры божьих коровок (то есть известно увеличение камеры и расстояние до снимаемого объекта);
2. Нас интересуют только божьи коровки, сидящие на листьях (значит, если принять, что листья зеленые, можно рассматривать только объекты, находящиеся на зеленом фоне);

Опираясь на перечисленные правила можно построить некий алгоритм их проверки и нахождения объектов на изображении, отвечающих этим правилам. Сложность заключается в том, что, во-первых, правила могут не описывать всех свойств объекта, во-вторых, правила могут выполняться не всегда, в-третьих, в процессе нахождения правил и их математической формулировке происходит ряд упрощений, уводя все дальше от вида реального объекта. Понятно, что успешность описанного метод напрямую зависит от фантазии и наблюдательности разработчика.

***Моделирование внешнего вида объекта, использование инструментария распознавания образов (pattern recognition) .***

Суть этого подхода заключается в вычислении некоторых числовых характеристик изображения моделируемого объекта (вектора признаков) и применение различных математических методов для определения "похожести" тестовых изображений на изображение объекта, основываясь на этих характеристиках.

Например, само изображение требуемого объекта можно напрямую представить как вектор в многомерном пространстве и натренировать некоторый классификатор с помощью набора примеров изображений объектов. Классификатор в данном случае означает некоторый инструмент, принимающий на вход изображение, представленное в виде вектора в многомерном пространстве, и выдающего на выходе некую информацию, классифицирующую входное изображение относительно некоторого признака.

Примеры часто используемых классификаторов:

1. Метод наименьших квадратов;
2. Прямое сравнение по какой-либо метрике пространства векторов признаков (например, сумме разности каждого элемента вектора) тестового изображения с изображениями-шаблонами (template-matching);
3. Нейросети (обычно для черно-белых изображений) – на входы нейросети подаются значения элементов вектора, на выходах формируется сигнал, классифицирующий объект на изображении;
4. Метод опорных векторов (support vector machines) – для распознавания изображений;
5. Моделирование многомерной функции распределения векторов признаков изображений объекта, оценка вероятности принадлежности тестового изображению к смоделированному распределению (факторный анализ, метод главных компонент, анализ независимых компонент, линейный дискриминантный анализ);
6. Деформируемые модели;

Прямое представление черно-белого изображения размера m\*n в качестве вектора порождает пространство размерности m\*n (яркость каждого пикселя – значение элемента вектора в таком пространстве). То есть изображение сравнительно небольшого разрешения (100x100) порождает пространство размерности 10,000. Работать в таком пространстве непросто, поэтому применяются различные методики снижения размерности, например метод главных компонент (principal components analysis, PCA)

Другие примеры характеристик (признаков) изображений, используемых для их классификации и распознавания:

1. Статистика распределения цветов (в различных представлениях, в том числе гистограмма изображения);
2. Статистические моменты (среднее, дисперсия, скрытые Марковские модели);

## Количество и свойства графических примитивов в объекте (прямых линий, окружностей – для распознавания символов) (на основе преобразования Хафа)

## 2.1. Метод наименьших квадратов

Перед тем, как начинать рассмотрение МГУА, было бы полезно вспомнить или узнать впервые метод наименьших квадратов — наиболее распространенный метод подстройки линейно зависимых параметров.

Рассмотрим для примера МНК для трех аргументов:

Пусть функция T=T(U, V, W) задана таблицей, то есть из опыта известны числа U­i, Vi, Wi, Ti ( i = 1, … , n). Будем искать зависимость между этими данными в виде:

(ф. 1)

где a, b, c — неизвестные параметры.

Подберем значения этих параметров так, чтобы была наименьшей сумма квадратов уклонений опытных данных Ti и теоретических Ti = aUi + bVi + cWi, то есть сумма:

(ф. 2)

Величина σ является функцией трех переменных a, b, c. Необходимым и достаточным условием существования минимума этой функции является равенство нулю частных производных функции σ по всем переменным, то есть:

(ф. 3)

Так как:

(ф. 4)

то система для нахождения a, b, c будет иметь вид:

(ф. 5)

Данная система решается любым стандартным методом решения систем линейных уравнений (Гаусса, Жордана, Зейделя, Крамера).

Рассмотрим некоторые практические примеры нахождения приближающих функций:

1. y = αx2 + βx + γ

Задача подбора коэффициентов α, β, γ сводится к решению общей задачи при T=y, U=x2, V=x, W=1, α=a, β=b, γ=c.

1. f(x, y) = αsin(x) + βcos(y) + γ/x

Задача подбора коэффициентов α, β, γ сводится к решению общей задачи при T=f, U=sin(x), V=cos(y), W=1/x, α=a, β=b, γ=c.

Если мы распространим МНК на случай с m параметрами,

(ф. 6)

то путем рассуждений, аналогичных приведенным выше, получим следующую систему линейных уравнений:

(ф. 7)

## где ,

## 2.2. Моделирование многомерной функции распреде

## ления векторов признаков изображений объекта

### 

### *Факторный Анализ(FA)*

Факторный анализ (ФА), как и многие методы анализа многомерных данных, опирается на гипотезу о том, что наблюдаемые переменные являются косвенными проявления относительно небольшого числа неких скрытых факторов. ФА, таким образом, это совокупность моделей и методов ориентированных на выявление и анализ скрытых (латентных)зависимостей между наблюдаемыми переменными. В контексте задач распознавания, наблюдаемыми переменными обычно являются признаки объектов.

Модели с латентными переменными применяются при решении следующих задач:

* понижение размерности признакового пространства,
* классификация объектов на основе сжатого признакового пространства,
* косвенной оценки признаков, не поддающихся непосредственному измерению,
* преобразование исходных переменных к более удобному для интерпретации виду.

Факторный анализ использует предположение о том, что исходные наблюдаемые переменные (распределенные по нормальному закону!) xi могут быть представлены в виде линейной комбинации факторов, также распределенных нормально:

xi=∑k=1..m(aikFk) + ui; i=1,...,n;

В этой модели присутствуют две категории факторов: общие факторы (common factors) Fk и специфические факторы(unique factors) ui. Фактор называется общим, если он оказывает влияние на две и более наблюдаемые переменные (математически это выражается в наличии как минимум двух существенно отличающихся от нуля коэффициентов aik для данного фактора Fk). Каждый из специфических факторов ui несет информацию только об одной переменной xi. Матрица aik называется матрицей факторных нагрузок (factor loadings) и задает влияние общих факторов на наблюдаемые переменные.

Содержательно, специфические факторы соответствуют необъясненной общими факторами изменчивости набора наблюдаемых переменных. Таким образом их можно рассматривать как случайную ошибку наблюдения или шум, не являющийся ценной информацией для выявления скрытых закономерностей и зависимостей. Важным предположением является независимость ui между собой. Обычно, однако не всегда, общие факторы Fk предполагаются некоррелированными (ортогональными).

Важными понятиями ФА являются общность и специфичность наблюдаемой переменной. На языке ФА доля дисперсии отдельной переменной, принадлежащая общим факторам (и разделяемая с другими переменными) называется общностью, дисперсия же приходящаяся на специфический фактор - специфичностью.

Целью ФА является выявление общих факторов Fk, специфических факторов ui и матрицы факторных нагрузок A таким образом, чтобы найденные общие факторы объясняли наблюдаемые данные наилучшим образом, то есть чтобы суммарная общность переменных была максимальна (а соответственно специфичность - минимальна).

Отличие Факторного Анализа (Factor Analysis, FA) от Метода Главных Компонент (Principal Components Analysis, PCA)

* Результатом ФА является модель, в явном виде описывающая зависимость наблюдаемых переменных от скрытых факторов (МГК это описательный анализ данных, без получения модели);
* ФА предусматривает ошибку моделирования (специфический фактор) для каждой из наблюдаемых переменных, в то время как МГК пытается объяснить всю изменчивость, включая шум, зависимостью от главных компонент;
* В МГК главные компоненты являются линейными комбинациями наблюдаемых переменных. В ФА наблюдаемые переменные являются линейными комбинациями общих и специфических факторов;
* Получаемые в результате ФА факторы могут быть использованы для интерпретации наблюдаемых данных;
* Главные компоненты некоррелированны (что эквивалентно их ортогональности при переносе начала координат в центр масс исходного набора), факторы же - не обязательно;
* МГК можно рассматривать как частный случай ФА, когда все специфические факторы приняты равными нулю, а общие факторы ортогональны.

### 

### *Метод главных компонент(PCA)*

Метод главных компонент применяется для снижения размерности пространства наблюдаемых векторов, не приводя к существенной потере информативности. Пусть дан исходный набор векторов линейного пространства Rn. Применение метода главных компонент позволяет перейти к базису пространства Rn, такому что:

Первая компонента (первый вектор базиса) соответствует направлению, вдоль которого дисперсия векторов исходного набора максимальна. Направление второй компоненты (второго вектора базиса) выбрано таким образом, чтобы дисперсия исходных векторов вдоль него была максимальной при условии ортогональности первому вектору базиса. Аналогично определяются остальные векторы базиса.

В результате, направления векторов базиса выбраны так, чтобы максимизировать дисперсию исходного набора вдоль первых компонент, называемых главными компонентами (или главными осями). Получается, что основная изменчивость векторов исходного набора векторов представлена несколькими первыми компонентами, и появляется возможность, отбросив оставшиеся (менее существенные) компоненты, перейти к пространству меньшей размерности.

Результатом применения МГК является вычисление матрицы W размера m x n, осуществляющей проекцию векторов пространства Rn на подпространство, натянутое на главные компоненты:

y = Wt(x - μ), y Rm, x Rn.



Где x - вектор из исходного набора, y - координаты вектора в подпространстве главных компонент, μ - средний вектор начального набора.

Главные компоненты (векторы базиса), выбираемые с помощью МГК, обладают следующим свойством: обратная проекция вектора y в Rn дает минимальную ошибку реконструкции (минимальное расстояние до образа вектора y). Нужно отметить, что корректное применение МГК возможно лишь при предположении о нормальном распределении векторов исходного набора.

В приложении к задаче классификации с учителем МГК обычно применяется следующим образом. После вычисления главных осей тренировочного набора, вектор признаков тестового объекта проецируется на подпространство, образованное главными осями. Вычисляются две характеристики: расстояние от проекции тестового вектора до среднего вектора тренировочного набора - Distance in Feature Space (DIFS), и расстояние от тестового вектора до его проекции в подпространство главных компонент - Distance From Feature Space (DFFS). Исходя из этих характеристик выносится решение о принадлежности тестового объекта классу, образованному тренировочным набором.

Отличие Факторного Анализа (Factor Analysis, FA) от Метода Главных Компонент (Principal Components Analysis, PCA)

* Результатом ФА является модель, в явном виде описывающая зависимость наблюдаемых переменных от скрытых факторов (МГК это описательный анализ данных, без получения модели);
* ФА предусматривает ошибку моделирования (специфический фактор) для каждой из наблюдаемых переменных, в то время как МГК пытается объяснить всю изменчивость, включая шум, зависимостью от главных компонент;
* В МГК главные компоненты являются линейными комбинациями наблюдаемых переменных. В ФА наблюдаемые переменные являются линейными комбинациями общих и специфических факторов;
* Получаемые в результате ФА факторы могут быть использованы для интерпретации наблюдаемых данных;
* Главные компоненты некоррелированы (что эквивалентно их ортогональности при переносе начала координат в центр масс исходного набора), факторы же - не обязательно;
* МГК можно рассматривать как частный случай ФА, когда все специфические факторы приняты равными нулю, а общие факторы ортогональны.

### 

### *Анализ независимых компонент(ICA). Начало формы*

### *Конец формы*

Задачей анализа независимых компонент (Independent Components Analysis, ICA) является разложение наблюдаемых случайных переменных xj в линейную комбинацию независимых случайных величин sk:

xj=aj1s1+aj2s2+...+ajnsn для всех j.

Основными предположениями, используемыми в данном методе, являются независимость компонент sk и, то, что их распределение отлично от нормального (non-gaussian). Алгоритм вычисления независимых компонент опирается на центральную предельную теорему, утверждающую, что при определенных условиях сумма независимо распределенных случайных величин стремится к нормальному распределению по мере увеличения количества слагаемых. Использую это утверждение, поиск независимых компонентов, как линейных комбинаций наблюдаемых переменных, ведется таким способом, чтобы получить независимые случайные величины, распределение которых максимально далеко от нормального. Степень близости распределения случайной величины к нормальному измеряется различным способами [Hyvarinen2000].

По своей формулировке, ICA близок к методу главных компонент (PCA) и факторному анализу (FA), однако имеет ряд существенных различий:

* В ICA существенно используется предположение о том, что распределения независимых компонент отличны от нормального,

что дает возможность интерпретировать ICA как FA для неортогональных факторов, с распределением отличным от нормального;

* В ICA понижение размерности не является целью, в отличии от FA и PCA;
* PCA добивается того, чтобы проекции векторов исходного набора на оси главных компонент были некоррелированы, в то время как ICA добивается их независимости (более сильное условие);
* Оси PCA ортогональны, в то время как оси независимых компонент - необязательно;

### 

### *Линейный Дискриминантный Анализ (Linear Discriminant*

### Analysis, LDA)

## Линейный Дискриминантный Анализ, в отличие от МГК и ФА не ставит своей целью найти подпространство меньшей размерности, наилучшим образом описывающее набор тренировочных изображений. Его задача - найти проекцию в пространство, в котором разница между различным классами объектов максимальна. Это требование формулируется как получение максимально компактных кластеров, соответствующих различным классам, удаленных на максимально возможное расстояние. С помощью ЛДА удается получить подпространство небольшой размерности, в котором кластеры изображений пересекаются минимально. Производить классификацию в таком пространстве значительно проще.

## 2.3. Деформируемые модели

Начало формы

Конец формы

В машинном зрении деформируемые модели являются мощным инструментом анализа и обработки данных. Деформируемые модели, в отличии от жестких (rigid), обладают большой гибкостью (имеют возможность представлять объекты с сильно различающейся формой) и в то же время дают возможность указать достаточно строгие ограничения на нежелательные изменения формы представляемых объектов.

В качестве примеров использования деформируемых моделей можно привести:

* выделение (локализация) объектов и структур определенного вида на 2D и 3D изображениях (черт человеческого лица, объектов на медицинских изображениях)
* отслеживания перемещения объектов между кадрами видеопотока.
* сегментация 2D и 3D изображений
* гладкая аппроксимации разреженного облака точек
* реконструкции 3D формы объекта по 2D изображениям - с помощью стерео, восстановления формы по закраске (shape from shading)

Использование деформируемых моделей при решение задачи обработки и распознавания изображений обычно позволяет в элегантной математической форме описать одновременное воздействие многих (возможно, противоречивых) факторов на процесс получения оптимального решения.

Конкретная деформируемая модель характеризуется:

* Способом задания формы модели (аналитические кривые и поверхности, конечные элементы);
* Способом измерения критерия согласия (goodness of fit) модели и измеренных данных;

## Способом модификации формы модели (по каким именно правилам (формулам) происходит изменение формы модели);

## 2.4. Скрытые Марковские Модели (Hidden Markov Models, HMM)

## Скрытые Марковские Модели (СММ) являются одним из способов получения математической модели (описания свойств) некоторого наблюдаемого сигнала. СММ относятся к классу стохастических моделей. Стохастические модели пытаются охарактеризовать только статистические свойства сигнала, не обладая информацией о его специфических свойствах. В основу стохастических моделей положено допущение о том, что сигнал может быть описан некоторым параметрическим случайным процессом и что параметры этого процесса могут быть достаточно точно оценены некоторым, вполне определенным способом. Настроенную СММ можно рассматривать как источник некоторого случайного сигнала со вполне определенными характеристиками. Для настроенной СММ есть возможность подсчитать вероятность генерации тестового сигнала данной моделью. В приложении к задаче распознавания, представив вектор признаков объекта в виде сигнала (набора последовательных наблюдений), можно смоделировать класс объектов с помощью СММ. Вероятность принадлежности тестового объекта классу, заданному СММ оценивается как вероятностью генерации сигнала, соответствующего его вектору признаков. Настройка (обучение) СММ - состоит в модификации ее параметров для того, чтобы добиться максимальной вероятности генерации сигналов, соответствующих векторам тренировочного набора.

## 2.5. Метод Опорных Векторов (Support Vector Machines, SVM)

Цель тренировки большинства классификаторов - минимизировать ошибку классификации на тренировочном наборе (называемую эмпирическим риском). В отличие от них, с помощью метода опорных векторов можно построить классификатор минимизирующий верхнюю оценку ожидаемой ошибки классификации (в том числе и для неизвестных объектов, не входивших в тренировочный набор). Применение метода опорных векторов к задаче распознавания заключается в поиске гиперплоскости в признаковом пространстве, отделяющей классы.

# Классификация с помощью опорных векторов позволяет использовать аппарат ядерных функций для неявного проецирования векторов-признаков в пространство потенциально намного более высокой размерности (еще выше, чем пространство изображений), в котором классы могут оказаться линейно разделимы. Неявное проецирование с помощью ядерных функций не приводит к усложнению вычислений, что позволяет успешно использовать линейный классификатор для линейно неразделимых классов.

# 3. Реализация программного обеспечения

В ходе работы над бакалаврской работой была написана программа, Выделяющая на изображении искомые объекты и нормирующая их.

Программа представляет собой простое SDI-приложение, написанное в среде Builder C++ v. 6.0. Для реализации обработки изображения использовалась библиотека Intel® Open Source Computer Vision Library (OpenCV).

В программе были использованы следующие функции библиотеки OpenCV:

|  |  |
| --- | --- |
| Функция | Описание функции |
| cvLoadImage | Загружает изображение из файла |
| cvSize | Возвращает размер изображения |
| cvCreateImage | Создает новое изображение |
| cvAbsDiff | Производит вычитание по модулю двух изображений |
| cvCmpS | Бинаризирует изображении, используя порог |
| cvErode | Выполняет побитовую операцию эрозии изображения |
| cvCamShift | Реализует алгоритм сравнения двух изображений |
| cvGetQuadrangleSubPix | Производит аффинное преобразование изображения |
| cvSetZero | Заполняет изображение черным цветом |
| cvFillConvexPoly | Заполняет многоугольник на изображении заданным цветом |
| cvCopy | Копирует изображение |

Также были использованы следующие классы:

|  |  |
| --- | --- |
| Класс | Описание класса |
| IplImage | Изображение |
| CvSize | Размер матрицы |
| CvPoint | Точка |
| CvRect | Прямоугольник |
| CvBox2D | Прямоугольник |
| CvMat | Матрица |

Алгоритм программы:

Начало → Загружаем изображение-эталон → Отображаем эталон на экране → Ждем щелчка мыши → Загружаем исследуемое изображение → Отображаем эталон на экране → Ждем щелчка мыши → Выполняем вычитание двух изображений → Бинаризируем результат → Отображаем результат на экране → Ждем щелчка мыши → Выполняем эрозию результата → Отображаем результат на экране → Ждем щелчка мыши → Находим минимальный прямоугольник, охватывающий найденные объекты → Поворачиваем исследуемое изображение → Отображаем результат на экране → Ждем щелчка мыши → Нормализуем результат → Отображаем результат на экране → дем щелчка мыши → Конец

**Заключение**

В работе создана программа обработки изображений. В дальнейшем планируется разработка программного обеспечения для классификации полученных образов.