

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН

Сәтбаев Университеті

Институт промышленной инженерии им. А. Буркитбаева

Кафедра «Робототехники и технических средств автоматизации»

Ғалым Махаббат Ғалмарқызы

Исследование особенностей регистрации визуальной информации в системе  
обработки изображений

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к дипломной работе

5B071600 – Приборостроение

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН

Сәтбаев Университеті

Институт промышленной инженерии им. А. Буркитбаева

Кафедра «Робототехники и технических средств автоматики»

**ДОПУЩЕН К ЗАЩИТЕ**  
Заведующий кафедрой РТиТСА  
к.т.н., профессор  
К.А. Ожигенов  
« 20 » 05 2019 г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
к дипломной работе

Тема: «Исследование особенностей регистрации визуальной информации в  
системе обработки изображений»

по специальности 5B071600 – Приборостроение

Выполнил

Ғалым Махаббат

Репензент  
кандидат технических наук,  
ассоц. проф

Научный руководитель  
кандидат физико-технических  
наук, доцент

Сейдилдаева А.К.  
« 20 » 05 2019 г.

Макешева К.К.  
« 24 » 05 2019 г.



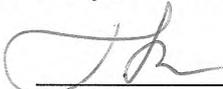
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ  
КАЗАХСТАН

Сәтбаев Университеті

Институт промышленной инженерии им. А. Буркитбаева

Кафедра «Робототехники и технических средств автоматики»

5B071600 – Приборостроение

**УТВЕРЖДАЮ**  
Заведующий кафедрой РТиТСА  
к.т.н., профессор  
К.А. Ожигенов  
  
« 20 » 05 2019 г.

**ЗАДАНИЕ**

**на выполнение дипломной работы**

Обучающемуся Галым Махаббат Галмаркызы

Тема: Исследование особенностей регистрации визуальной информации в системе обработки изображений

Утверждена приказом Ректора Университета № 125-п от «06» 11 2019 г.

Срок сдачи законченной работы «24» ноя 2019 г.

Исходные данные к дипломной работе: интегрированная среда разработки для языка программирования Python

Перечень подлежащих разработке вопросов в дипломной работе:

- а) процессы получения визуальной информации человеком;
- б) процессы получения визуальной информации системой компьютерного зрения;
- в) получить визуализацию автоматизированного процесса регистрации визуальной информации;

Перечень графического материала (с точным указанием обязательных чертежей):

представлены 15 слайдов презентации работы

Рекомендуемая основная литература: из 20 наименований 16

**ГРАФИК**  
подготовки дипломной проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Технологическая часть	15.01 – 05.02.2019 г.	<i>Вопышено</i>
Программная часть	02.03 – 20.04.2019 г.	<i>Вопышено</i>

**Подписи**  
консультантов и нормоконтролера на законченный проект  
с указанием относящихся к ним разделов проекта

Наименование разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Нормконтролер	Ж.С.Бигалиева, магистр технических наук, лектор	20.05.2019 г.	<i>[Подпись]</i>

Научный руководитель *Макешева* Макешева К. К.

Задание принял к исполнению обучающийся *Г. М. Г.* Галым М.Г.

Дата « 24 » 05 2019 г.

## АҢДАТПА

Дипломдық жұмыста нақты уақыт режимінде объектілерді тану үшін алгоритм жасалды. Дипломдық жұмыс үш бөліктен тұрады: адам және компьютермен визуалды ақпаратты алу процесінің жалпы көрінісі, оптикалық кескінді цифрлық өңдеу және OpenCV ашық кітапханасы.

Бірінші бөлім адам мен компьютер арқылы көрнекі ақпарат алу процесіне шолуды қамтамасыз етеді.

Екінші бөлім ақпарат алудың негізгі ұғымдары мен негізгі бөліктерін сипаттайды.

Үшінші бөлім алгоритмді жазбаша түрде қолданып, алгоритмді талдау үшін пайдаланылатын кітапхананы сипаттайды.

## АННОТАЦИЯ

В данной дипломной работе был разработан алгоритм для распознавания объектов в реальном времени. Дипломная работа состоит из трех частей: общее представление процесса получения визуальной информации человеком и компьютером, цифровая обработка оптического изображения и открытая библиотека OpenCV.

В первой части приведено общее представление процесса получения визуальной информации человеком и компьютером.

Во второй части описаны основные понятия и основные части процесса получения информации.

В третьей части описана библиотека, которая была использована в написании алгоритма и разбор самого алгоритма.

## ANNOTATION

In this diploma work, an algorithm was developed for recognizing objects in real time. The diploma work consists of three parts: a general view of the process of obtaining visual information by a person and a computer, digital optical image processing and an open library OpenCV.

The first part provides an overview of the process of obtaining visual information by a person and a computer.

The second part describes the basic concepts and basic parts of the process of obtaining information.

The third part describes the library that was used in writing the algorithm and parsing the algorithm itself.

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ	9
1 Процесс получения визуальной информации человеком и системой компьютерного зрения	10
2 Основы регистрации и цифровой обработки оптического изображения	11
2.1 Основные понятия компьютерной обработки изображений	11
2.1.1 Цифровое изображение	11
2.1.2 Методы обработки изображения	12
2.2 Основные понятия теории сигналов	13
2.2.1 Преобразование сигналов оптическими системами	13
2.2.2 Преобразование сигналов оптическими системами	15
2.2.3 Частотные передаточные характеристики	15
2.3 Дискретизация и квантование непрерывных изображений	17
2.3.1 Дискретизация непрерывных изображений	17
2.3.2 Квантование изображений	18
2.4 Ввод изображений	19
2.4.1 Глубина цвета	21
3 OpenCv свободная библиотека компьютерного зрения	23
3.1 Распознавание объектов с web камеры в реальном времени	23
ЗАКЛЮЧЕНИЕ	27
СПИСОК ИСПОЛЬЗОВАННЫХ ЛИТЕРАТУР	28
ПРИЛОЖЕНИЕ А	

## ВВЕДЕНИЕ

В настоящее время цифровая обработка и последующее распознавание изображений - одно из интенсивно развивающихся направлений научных исследований. 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.

Актуальность дипломной работы заключается в разработке алгоритма для распознавания объектов в реальном времени с помощью открытой библиотеки OpenCV и imutils.

Цель работы. Целью дипломной работы является исследование основ моделирования, регистрации и цифровой обработки оптического изображения и на этой основе создание алгоритма обработки изображений в реальном времени.

Дипломная работа состоит из трех глав, заключения и списка использованной литературы.

# 1 ПРОЦЕСС ПОЛУЧЕНИЯ ВИЗУАЛЬНОЙ ИНФОРМАЦИИ ЧЕЛОВЕКОМ И СИСТЕМОЙ КОМПЬЮТЕРНОГО ЗРЕНИЯ

Человеческий глаз – хрупкий прибор, предназначенный для получения и обработки всей световой информации, поступающей в глаз. У человека формирование картинки осуществляется на сетчатке с последующей обработкой и записыванием ее в память в мозгу.



Рисунок 1.1 - Процесс получения визуальной информации человеком

Свет состоит из спектрального распределения электромагнитной энергии с длинами волн в диапазоне 400 - 700 нм. За пределами этого диапазона, волны называются ультрафиолетовыми (УФ) и инфракрасными (ИК). С восприятия начинается обработка человеком графической, видео- и аудиоинформации. Согласно теории цветового зрения, цвет воспринимается рецепторами, светочувствительной сетчатой оболочкой глаза палочками и колбочками способными воспринимать свет различных длин волн.

Системы компьютерного зрения во многом напоминают этапы обработки визуальной информации человеком, но в зависимости от назначения системы и области её применения, имеют существенные различия и свои особенности.



Рисунок 1.2 - Процессы обработки визуальной информации человеком и компьютером

## 2 ОСНОВЫ РЕГИСТРАЦИИ И ЦИФРОВОЙ ОБРАБОТКИ ОПТИЧЕСКОГО ИЗОБРАЖЕНИЯ

### 2.1 Основные понятия компьютерной обработки изображений

При рассмотрении вопросов, связанных с обработкой изображений, необходимо определить само понятие «изображение». Самое обычное и широкое обозначение этого понятия: изображение – это то, что мы видим. Иное определение: изображение – это информация, для визуального восприятия. В зависимости от происхождения относительно можно выделить следующие типы изображений:

- нарисованное или печатное (художник, полиграфия, принтер);
- оптическое (распределение интенсивности электромагнитного поля создаваемое оптическим прибором в некоторой области пространства (области локализации) например, на сетчатке глаза, на экране при проецировании, в плоскости приемника объектива фотоаппарата);
- фотографическое (оптическое изображение, зарегистрированное на фотоматериале);
- электронное или цифровое (оптическое изображение, зарегистрированное с помощью электронного приемника). Электронным обычно называют изображение, отображаемое на экране монитора.

Очевидно, что данное деление условное, так как изображение из одного типа сразу переходит в другой. Надо дать изображению формализованное обозначение, которое дает нам возможность описать этот объект математически и управлять им для достижения определенных целей. Это принято называть «обработка изображений».

Обработка изображений может производиться в различных целях:

- Изменение изображения с целью достижения определенных эффектов (художественное улучшение).
- Image Processing – визуальное улучшение качества изображения (коррекция яркости и контраста, цветокоррекция и т.п.); объективное улучшение качества изображения (устранение искажений, типадисторсия, смаз, расфокусировка и т.п.);
- Image Analysis – проведение измерений на изображении (анализ интерферограмм, гартманнограмм и т.п.);
- Image Understanding – распознавание образов (распознавание символов, отпечатков пальцев, лиц, приборы наведения и т.п.).

#### 2.1.1 Цифровое изображение

Изображение – это функция двух вещественных переменных  $I(x, y)$ , где  $I$  – это интенсивность в точке с координатами  $(x, y)$ . В некоторых случаях к обработке подвергается не полностью изображение, а только его часть которая называется region-of-interest (область представляющая интерес).

Чтобы обработать изображение на компьютере изображение должно быть дискретизировано и квантовано. Это изображение называется цифровым.

Цифровое изображение  $A(m, n)$  представлено в дискретном двумерном пространстве, где  $m$  – номер строки, а  $n$  – номер столбца. Элемент, расположенный на пересечении  $m$ -ой строки и  $n$ -го столбца называется пиксел (pixel – picture element). Яркость пиксела может передаваться либо вещественным или целым числом. Касательная интенсивность в вещественных числах изменяется от 0 до 1, а в целых числах от 0 до 255.

Физический сигнал, возникающий в точке  $(x, y)$ , является функцией, которая зависит от многих параметров ( $z$  – глубина,  $\lambda$  – длина волны,  $t$  – время).

Среди характеристик цифровых изображений следует выделить:

Размер: этот параметр может быть любым, но часто выбирается исходя из особенностей регистрации изображения (например, видеостандарты PAL (625, 4:3), SECAM (625, 4:3), NTSC (525, 4:3)), особенностей последующей обработки (алгоритмы быстрого преобразования Фурье предъявляют особые требования) и т.п.

Количество цветов (глубина цвета): точнее количество бит, отводимое для хранения цвета, определяется упрощением электронных схем и кратно степени 2. Изображение для хранения информации о цветах которого необходим 1 бит называется бинарным. Для хранения полутоновых (gray scale, gray level) изображений используется обычно 8 бит. Цветные изображения хранятся обычно с использованием 24 бит по 8 на каждый из трех цветовых каналов.

Разрешение: измеряется в dpi (dot per inch – число точек на дюйм). К примеру, на экране монитора разрешение обычно 72 dpi, при выводе на бумагу – 600 dpi, при регистрации на ПЗС-матрице с размером одного элемента 9 мкм разрешение составит почти 3000 dpi. В процессе обработки разрешение можно изменить: на само изображение это не повлияет, но изменится его отображение устройством визуализации.

### 2.2.2 Методы обработки изображения

В основе алгоритмов обработки изображений положены в основном интегральные преобразования: свертка, преобразование Фурье и др. Также используются статистические методы.

Методы обработки изображений группируют по количеству пикселей участвующих в одном шаге преобразования:

- поточечные методы: преобразование значения в точке  $a(m, n)$  в значение  $b(m, n)$  независимо от соседних точек;
- локальные (окрестностные) методы: используют значения соседних точек в окрестности  $a(m, n)$  чтобы вычислить значение  $b(m, n)$ ;

– глобальные методы: на основе всех значений исходного изображения  $A(m, n)$  определяют значение  $b(m, n)$ .

## 2.2 Основные понятия теории сигналов

Прежде всего, следует определить само понятие, как сигнал. В общем случае сигнал – это изменение некоторой физической величины. В зависимости от области определения говорят о временной, частотной или пространственной форме представления сигнала.

Чаще всего сигналы рассматривают как функцию, заданную в некоторых физических координатах. Математической моделью сигнала чаще всего являются скалярные функции. Но иногда используются более сложные модели. К примеру, для описания электромагнитного поля удобно использовать комплексные функции, а для цветных изображений – трехкомпонентные векторные функции.

Если область применения сигнала непрерывна, то он называется непрерывным или аналоговым. Название "аналоговый" дано потому, что они являются "аналогами" реальных физических процессов, происходящих в действительности. Такие сигналы могут принимать любые значения. Можно привести в пример аналогового сигнала изменение напряжения.

Сигнал, аргументы которого принимают счетное множество значений, называется дискретным. Если же сам сигнал принимает счетное множество значений, то он называется квантованным. Цифровыми называются дискретные квантованные сигналы.

### 2.2.1 Преобразование сигналов оптическими системами

Сигналы в оптических системах подвергаются разнообразным преобразованиям. Для математического описания этих преобразований в общем случае необходимо задать все возможные пары входных и выходных сигналов. Однако объем такого описания настолько велик, что фактически исключает возможность его практического использования. Поэтому модели оптических систем и описания преобразования сигналов в них строятся по иерархическому принципу. При этом преобразование сигналов представляются как совокупность некоторых элементарных преобразований. Оптический прибор при этом рассматривается как каскад преобразователей информации, а оптическая система является линейным фильтром сигнала.

Каждый изображающий прибор принимает информацию от предыдущего элемента каскада и передает последующему. Входной сигнал называют предметом, а выходной – изображением. При построении модели абстрагируемся от конкретного физического содержания предмета и изображения и будем рассматривать их как некоторые обобщенные

сигналы или функции  $I(x)$  и  $I'(x')$  обобщенных интенсивностей от векторов обобщенных координат  $x = \frac{x}{y}$  и  $x' = \frac{x'}{y'}$ .

Задачей изображающего прибора является преобразование входного сигнала – функции предмета  $I(x)$  в выходной сигнал – функцию изображения  $I'(x')$ . Модель оптического прибора, описывающая общие закономерности формирования изображения в оптических системах, не связанные с физическими принципами их работы (внешняя функциональная модель), есть оператор  $L$ , осуществляющий преобразование:

$$I'(x') = L[I(x)] \text{ или } I(x) \xrightarrow{L} I'(x') \quad (1)$$

В теории изображения предполагается, во-первых, что этот оператор должен удовлетворять условию линейности. Линейные преобразования – это преобразования, для которых выполняется принцип суперпозиции. Математически это записывается следующим образом: преобразование  $L$  является линейным, если для любых сигналов  $I_k$  заданных в линейном пространстве, и скаляров  $a_k$  справедливо:

$$L[\sum_k a_k I_k] = \sum_k a_k L[I_k] \quad (2)$$

В соответствии с этим выражением, изображение суммы равно сумме изображений.

Система, осуществляющая линейные преобразования, называется линейной. Для таких систем наиболее распространенным является описание с помощью импульсной реакции, определяемой как отклик оператора на дельта-функцию:

$$h(x') = L[\delta(x)] \quad (3)$$

В теории оптических изображающих систем эта импульсная функция называется функцией рассеяния точки (ФРТ), и представляет собой изображение (пятно рассеяния) светящейся точки единичной энергии.

Во-вторых, для изображающего оператора  $L$  должно выполняться условие изопланатичности или пространственной инвариантности (инвариантности к сдвигу):

$$I(x - a) = L[I'(x' - a')] \quad (4)$$

В соответствии с этим выражением, при смещении предмета на вектор  $a$  изображение только смещается на вектор  $a'$ , причем  $a'$  пропорционален  $a$ , а именно:  $a' = Va$ , где  $V$  – матрица обобщенных увеличений.

## 2.2.2 Структурные передаточные характеристики

Изображающие оптические системы могут давать изображения различного качества в плане передачи тонкой структуры предмета. При рассмотрении передачи структуры предмета используют нормированную ФРТ, энергия которой равна единице:

$$h_H(x') = \frac{h(x')}{H}; \iint_{-\infty}^{+\infty} h_H(x') dx' = 1 \quad (5)$$

где  $H = \frac{E'}{E}$  – передняя зональная обобщенная светосила,  $E = \iint_S I(x) dx$  и  $E' = \iint_S I'(x') dx'$  – обобщенная энергия участка предмета и изображения соответственно.

Нормированная ФРТ характеризует в чистом виде передачу системой структуры предмета.

В дальнейшем для упрощения будем опускать в обозначении индекс  $H$  и под  $h(x')$  подразумевать нормированную ФРТ.

Рассмотрим теперь предмет и изображение приведенными на одну поверхность, то есть изображаемыми с единичным увеличением. Благодаря этому исключаются масштабные преобразования, описываемые матрицей обобщенных увеличений. Передача структуры предмета в таком случае будет описываться следующим выражением:

$$I(x') \iint_{-\infty}^{+\infty} I(x) h(x' - x) dx \quad (6)$$

где координаты  $x$  и  $x'$  рассматриваются в одном масштабе. В математике подобное выражение называется сверткой и записывается следующим образом:

$$I' = I \otimes h \quad (7)$$

Таким образом, передача структуры предмета описывается сверткой функции предмета с нормированной функцией рассеяния точки.

## 2.2.3 Частотные передаточные характеристики

Описание передачи структуры предмета в виде приведенного выше выражения свертки является полным, но не всегда удобно и наглядно. Это связано в первую очередь с тем, что сами функции предмета  $I(x)$  и изображения  $I'(x')$  не вполне наглядно представляют тонкую структуру предмета и изображения, и должны быть заменены другими.

Наиболее подходящим эталоном структуры является периодический гармонический объект. Он характеризуется пространственной частотой  $\nu$

(величиной, обратной периоду  $T$ ), углом ориентации  $\theta$ , амплитудой  $u$  и начальным сдвигом  $b$  (начальной фазой  $\varphi = 2\pi\nu b$ ). Причем  $u$  и  $b$  могут быть объединены в одну комплексную величину – комплексную амплитуду  $g = ue^{-i\varphi}$ . Чем больше пространственная частота  $\nu$ , тем тоньше структура предмета. Пространственная частота имеет единицы измерения, обратные обобщенным координатам.

Теперь перейдем к такому важному понятию, как спектр. В общем случае спектром называется совокупность значений какой-либо величины, характеризующей систему или процесс.

Структурное содержание сложного объекта  $I(x)$  или изображения  $I'(x')$  нагляднее описывается спектрами пространственных частот  $\dot{I}(\nu)$  и  $\dot{I}'(\nu')$ , показывающих распределение комплексных амплитуд по пространственным колебаниям, на которые могут быть разложены  $I(x)$  и  $I'(x')$ . Функции предмета и изображения и их спектры частот связаны между собой преобразованием Фурье в соответствии со следующими соотношениями:

$$I(x) \overset{F}{\leftrightarrow} \dot{I}(\nu) \text{ или } \dot{I}(\nu) = \iint_{-\infty}^{+\infty} I(x)e^{2\pi i(\nu^T x)} dx, \quad (8)$$

$$\dot{I}'(x') \overset{F}{\leftrightarrow} \dot{I}'(\nu') \text{ или } \dot{I}'(\nu') = \iint_{-\infty}^{+\infty} I'(x')e^{2\pi i(\nu'^T x')} dx' \quad (9)$$

Таким образом, передача изображающей системой структуры предмета нагляднее всего описывается как передача его спектра пространственных частот. Если теперь применить преобразование Фурье к обеим частям выражения свертки, описывающего передачу структуры предмета, учитывая свойства преобразования Фурье, получаем:

$$\dot{I}'(\nu') = \dot{I}(\nu)D(\nu) \quad (10)$$

где функция  $D(\nu) = F[h(x')]$  представляет собой двумерное преобразование Фурье от ФРТ и называется оптической передаточной функцией (ОПФ). Приведенное выражение называется соотношением фильтрации и показывает, что спектр пространственных частот изображения получается как произведение спектра пространственных частот предмета на оптическую передаточную функцию системы. Эта функция служит наиболее наглядной и удобной структурной (частотной) передаточной характеристикой.

ОПФ в общем случае является комплексной функцией:

$$D(\nu) = T(\nu)e^{i\varphi(\nu)} \quad (11)$$

где её модуль  $T(\nu)$  – частотно-контрастная характеристика (ЧКХ) или модуляционная передаточная функция (МПФ), а её аргумент  $\varphi(\nu)$

– частотно-фазовая характеристика (ЧФХ) или фазовая передаточная функция (ФПФ).

## 2.3 Дискретизация и квантование непрерывных изображений

В систему обработки информации сигналы поступают, как правило, в непрерывном виде. Для компьютерной обработки непрерывных сигналов необходимо, прежде всего, преобразовать их в цифровые. Для этого выполняются операции дискретизации и квантования.

### 2.3.1 Дискретизация непрерывных изображений

Дискретизация – это преобразование непрерывного сигнала в последовательность чисел (отсчетов), то есть представление этого сигнала по какому-либо конечномерному базису. Это представление состоит в проектировании сигнала на данный базис.

Наиболее удобным с точки зрения организации обработки и естественным способом дискретизации является представление сигналов в виде выборки их значений (отсчетов) в отдельных, регулярно расположенных точках. Такой способ называют растриванием, а последовательность узлов, в которых берутся отсчеты – растром. Интервал, через который берутся значения непрерывного сигнала называется шагом дискретизации. Обратная шагу величина называется частотой дискретизации.

Существенный вопрос, возникающий в ходе дискретизации: с какой частотой брать отсчеты сигнала для того, чтобы была возможность его обратного восстановления по этим отсчетам? Очевидно, что если брать отсчеты слишком редко, то в них не будет содержаться информация о быстро меняющемся сигнале. Скорость изменения сигнала характеризуется верхней частотой его спектра. Таким образом, минимально допустимая ширина интервала дискретизации связана с наибольшей частотой спектра сигнала (обратно пропорциональна ей).

Для случая равномерной дискретизации справедлива теорема Котельникова, опубликованная в 1933 году в работе “О пропускной способности эфира и проволоки в электросвязи”. Она гласит: если непрерывный сигнал  $x(t)$  имеет спектр, ограниченный частотой  $F_{\max}$  то он может быть полностью и однозначно восстановлен по его дискретным отсчетам, взятым с периодом  $T = \frac{1}{2F_{\max}}$ , т.е. с частотой  $f_d = 2 F_{\max}$ .

Восстановление сигнала осуществляется при помощи функции  $\text{sinc}(x) = \frac{\sin(x)}{x}$ . Котельниковым было доказано, что непрерывный сигнал, удовлетворяющий приведенным выше критериям, может быть представлен в виде ряда:

$$x(t) = \sum_{k \in \mathbb{Z}} x(kT) \operatorname{sinc}\left(\frac{t}{T} - k\right) \quad (12)$$

Эта теорема так же еще называется теоремой отсчетов. Функция  $\operatorname{sinc}(x)$  называется еще функцией отсчетов или Котельникова, хотя интерполяционный ряд такого вида изучал еще Уитакер в 1915 году. Функция отсчетов имеет бесконечную протяженность по времени и достигает наибольшего значения, равного единице, в точке  $k = \frac{t}{T}$ , относительно которой она симметрична.

Каждую из этих функций можно рассматривать как отклик идеального фильтра низких частот (ФНЧ) на дельта-импульс, пришедший в момент времени  $kT$ . Таким образом, для восстановления непрерывного сигнала из его дискретных отсчетов, их необходимо пропустить через соответствующий ФНЧ. Следует заметить, что такой фильтр является некаузальным и физически нереализуемым.

Приведенное соотношение означает возможность точного восстановления сигналов с ограниченным спектром по последовательности их отсчетов. Сигналы с ограниченным спектром – это сигналы, спектр Фурье которых отличен от нуля только в пределах ограниченного участка области определения. Оптические сигналы можно отнести к ним, т.к. спектр Фурье изображений, получаемых в оптических системах, ограничен из-за ограниченности размеров их элементов. Частоту  $F_{\max}$  называют частотой Найквиста. Это предельная частота, выше которой во входном сигнале не должно быть спектральных компонентов.

### 2.3.2 Квантование изображений

При цифровой обработке изображений непрерывный динамический диапазон значений яркости делится на ряд дискретных уровней. Эта процедура называется квантованием. Её суть заключается в преобразовании непрерывной переменной  $x$  в дискретную переменную  $x_{\text{кв}}$ , принимающую конечное множество значений  $\{r_1, \dots, r_L\}$ . Эти значения называются уровнями квантования. В общем случае преобразование выражается ступенчатой функцией (рис. 1). Если интенсивность  $x$  отсчета изображения принадлежит интервалу  $[d_j, d_{j+1}]$  (т.е., когда  $d_j < x \leq d_{j+1}$ ), то исходный отсчет заменяется на уровень квантования  $r_j$ , где  $d_j, j=1, L+1$  – пороги квантования. При этом полагается, что динамический диапазон значений яркости ограничен и равен  $[d_j, d_{L+1}]$ .

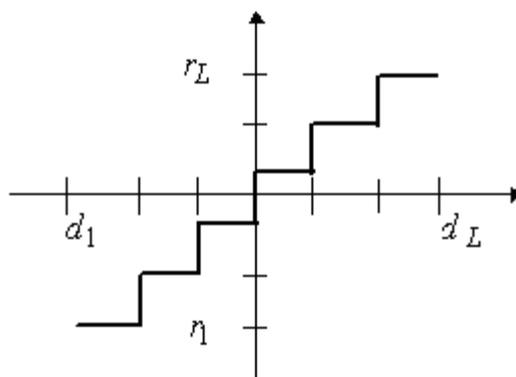


Рисунок 2.3.1. - Функция, описывающая квантование

Основная задача при этом состоит в определении значений порогов  $d_j$  и уровней  $r_j$  квантования. Простейший способ решения этой задачи состоит в разбиении динамического диапазона на одинаковые интервалы. Однако такое решение не является наилучшим. Если значения интенсивности большинства отсчетов изображения сгруппированы, например, в "темной" области и число уровней  $L$  ограничено, то целесообразно квантовать неравномерно. В "темной" области следует квантовать чаще, а в "светлой" реже. Это позволит уменьшить ошибку квантования.

В системах цифровой обработки изображений стремятся уменьшить число уровней и порогов квантования, так как от их количества зависит объем информации, необходимый для кодирования изображения. Однако при относительно небольшом числе уровней на квантованном изображении возможно появление ложных контуров. Они возникают вследствие скачкообразного изменения яркости проквантованного изображения и особенно заметны на пологих участках ее изменения. Ложные контуры значительно ухудшают визуальное качество изображения, так как зрение человека особенно чувствительно именно к контурам. При равномерном квантовании типичных изображений требуется не менее 64 уровней.

## 2.4 Ввод изображений

Как уже было сказано выше, оптическое изображение с точки зрения теории сигналов является двумерным непрерывным сигналом. В таком виде оно не пригодно для обработки в компьютерных системах, и, следовательно, должно быть преобразовано. Для этого выполняются операции дискретизации (по пространственным координатам) и квантования (по интенсивности). Получившееся цифровое изображение представляет собой массив дискретных отсчетов (последовательность цифровых значений). Устройство, выполняющее операции дискретизации и квантования, в теории сигналов называют АЦП (аналого-цифровой преобразователь), а выполняющее обратное преобразование – ЦАП (цифро-аналоговый преобразователь).

Все компьютерные изображения, форматы их представления и программы создания и редактирования делят на два больших класса – векторные и растровые.

Векторные изображения представляют собой набор математически описанных объектов. Такой подход применяется при создании и хранении чертежной документации, планов, схем, диаграмм и т.п.

Изображение, представленное в цифровом виде, имеет вид прямоугольной матрицы (растра, растровой или битовой карты, bitmap), состоящей из набора дискретных элементов. Каждый такой элемент растра носит название пиксел (от английского pixel – picture element). Графические изображения, представленные в таком виде, получили название растровых.

Количество элементов (пикселей) на единицу длины называется – разрешением. Оно измеряется в большинстве случаев в точках на дюйм (dpi, сокращенное от dot per inch) или пиксел на дюйм (ppi, сокращенное от pixel per inch). По сути оба эти понятия обозначают одно и то же. Разница лишь в том, что в первом случае единичный элемент изображения назван точкой (dot), а во втором – пикселем (pixel). Разрешение цифровых изображений – понятие неоднозначное, поскольку каждая стадия процесса воспроизведения накладывает свои требования и ограничения. Рассмотрим этапы последовательно.

Одним из наиболее распространенных устройств предназначенных для оцифровки изображений является сканер. Сканер – это устройства ввода текстовой или графической информации в компьютер путем преобразования ее в цифровой вид для последующего использования, обработки, хранения или вывода. Таким образом, на этапе сканирования осуществляется перевод изображение из аналоговой формы в цифровую. Разрешение, задаваемое на этом этапе, определяет какое количество пикселей будет получено на один дюйм исходного изображения. Разрешение – один из важнейших параметров сканера. Оно бывает физическое и интерполяционное. Первое зависит от конструкции устройства и в ряде случаев может быть переменным. Практически во всех моделях сканеров (особенно недорогих) существует и второй тип разрешения – интерполяционное. Суть его в том, что на некотором участке по имеющимся цифровым данным полиномом необходимой степени воспроизводится функция, в приближении отражающая существовавший аналоговый сигнал. Затем по этой функции производится перевыборка (изменение шага дискретизации). Таким образом, можно искусственно повысить реальное разрешение сканера.

На этапе выполнений преобразований цифрового изображения в компьютере понятие разрешающей способности в достаточной мере условно. Фактически оно определяет, какого размера будет изображение в случае его вывода (например, на печать или экран). Ни на какие цифровые преобразования разрешение не влияет. Все цифровые преобразования производятся над пикселями, поэтому на этапе компьютерной обработки, разрешения не играет никакой роли.

На этапе вывода может применяться большое количество разнообразных устройств. Здесь под разрешением понимают количество точек, которое может вывести то или иное устройство на единицу длины. Понятно, что при этом один пиксел изображения в цифровом виде может быть не равен одному пикселю устройства вывода.

Для того чтобы определить, какое разрешение устройство ввода должно быть выбрано для сохранения структуры исходного изображения воспользуемся теоремой Котельникова. Согласно ей, для передачи определенной пространственной частоты (например, 100 лин/мм) потребуется сканирование с вдвое большей частотой (200 лин/мм). И действительно, если при сканировании миры, имеющей шаг 100 лин/мм, используется разрешение так же в 100 линий на мм, то ее структура будет передана полностью в том случае, если каждая линия миры попадет на каждую линию сканирующего устройства. Если же линии миры окажутся между сканирующими линиями, то получим серое поле. Пространственная частота сканера в 200 лин/мм означает, что имеется 400 переходов черное/белое или 400 считывающих элементов на мм. Таким образом получаем, что при сканировании миры с частотой 100 лин/мм необходимо иметь разрешение сканирующего устройства как минимум 400 точек на мм (ppm).

#### 2.4.1 Глубина цвета

Цвет каждого пиксела цифрового изображения описывается несколькими числами (в зависимости от используемой цветовой системы). Количество бит, отводимое на представление информации о цвете каждого пиксела, называют глубиной цвета (color depth) или битовой глубиной цвета (bit depth). Иногда под цветовой глубиной понимают максимальное количество цветов, которые можно представить.

Глубина цвета определяет, как много цветов может быть использовано при отображении одного пиксела. Например, если цветовая глубина равна 1 бит, то пиксел может представлять только один из двух возможных цветов – белый или черный. Если цветовая глубина равна 8 бит, то количество возможных цветов равно  $2^8 = 256$ . При глубине цвета 24 бит количество цветов превышает 16 миллионов, что фактически превосходит способность глаза человека разрешать цвета. Такой режим называется True Color (истинный цвет). В связи с тем, что 24-разрядное представление неудобно с точки зрения обработки изображения, обычно в режиме TrueColor используется 32 бита. В случае 32-разрядного представления информации о цвете младшие три байта по-прежнему описывают цвет точки, а старший байт либо управляет дополнительными параметрами (например, альфа-каналом, информацией о взаимном перекрывании объектов или глубине в трехмерном изображении), либо не используется. Понятно, что при таком представлении увеличивается размер изображения, однако существенно возрастает скорость его обработки центральным и графическим процессорами компьютера.

## 3 OpenCV СВОБОДНАЯ БИБЛИОТЕКА КОМПЬЮТЕРНОГО ЗРЕНИЯ

OpenCV (Open Source Computer Vision Library) представляет собой библиотеку программного обеспечения для компьютерного зрения с открытым исходным кодом и машинного обучения.

OpenCV создан для обеспечения общей инфраструктуры приложений основанных на компьютерном зрении для упрощения использования наработок и решений большого количества разработчиков в готовых продуктах.

Поскольку OpenCV распространяется под BSD лицензией, любой желающий может использоваться её как в исследовательских целях, так и в коммерческих.

В библиотеке присутствует более 2500 оптимизированных алгоритмов, которые включают в себя полный набор классических и современных алгоритмов компьютерного зрения и машинного обучения.

Эти алгоритмы могут использоваться для обнаружения и распознавания лиц, идентификации объектов, классификации действий человека в видео, отслеживания движения камеры, отслеживания движущихся объектов, извлечения 3D-моделей объектов, создания 3D-облаков точек из стереокамер, сшивания изображений вместе для получения высокого разрешения изображение всей сцены, найти похожие изображения из базы данных изображений, удалять красные глаза с изображений, сделанных с помощью вспышки, следить за движениями глаз, распознавать декорации и устанавливать маркеры, чтобы накладывать их на дополненную реальность и т. д.

Библиотека широко используется в компаниях, исследовательских группах и правительственных органах.

### 3.1 Распознавание объектов с web камеры в реальном времени

Чтобы написать программу для распознавания объектов в реальном времени мы должны первым делом получить доступ к нашей web камере и применить распознавание объекта для каждого кадра.

Для начала создаем файл и называем его `real_time_object_detection.py` и импортируем туда библиотеки `imutils` и `OpenCV`.

```
# импортировать необходимые пакеты
from imutils.video import VideoStream
from imutils.video import FPS
import numpy as np
import argparse
import imutils
import time
import cv2
```

Далее анализируем аргументы командной строки:

```

# анализируем аргументы командной строки
ap = argparse.ArgumentParser()
ap.add_argument("-p", "--prototxt", required=True,
                help="path to Caffe 'deploy' prototxt file")
ap.add_argument("-m", "--model", required=True,
                help="path to Caffe pre-trained model")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
                help="minimum probability to filter weak detections")
args = vars(ap.parse_args())

```

- prototxt : Путь к prototxt Caffe файлу.
- model : Путь к предварительно подготовленной модели.
- confidence : Минимальный порог валидности (сходства) для распознавания объекта (значение по умолчанию — 20%).

Добавляем основные объекты и затем мы инициализируем список классов и набор цветов которые будут ограничивать каждый класс:

```

# инициализировать список меток классов SSD MobileNet был обучен
обнаружению
# затем сгенерировать набор цветов ограничивающего прямоугольника для каждого
# класса
CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
           "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
           "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
           "sofa", "train", "tvmonitor"]
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))

```

Дальше загружаем модель и настраиваем наш видео поток:

```

# инициализировать видеопоток,
# и инициализировать счетчик FPS
print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
time.sleep(2.0)
fps = FPS().start()

```

Загружаем нашу сериализованную модель, предоставляя ссылки на prototxt и модели. Затем инициализируем видео поток (веб-камера). Сначала запускаем VideoStream, затем мы ждём, пока камера включится, и наконец начинаем отсчёт кадров в секунду. Классы VideoStream и FPS являются частью пакета imutils.

Теперь проходим по каждому кадру:

```

# цикл по кадрам из видеопотока
while True:
    # возьмите кадр из видеопотока
    # и измените его размер до максимальной ширины 400 пикселей
    frame = vs.read()
    frame = imutils.resize(frame, width=400)

    # захватить размеры кадра и преобразовать его в blob
    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)),

```

```

0.007843, (300, 300), 127.5)

# передать blob через сеть и получить обнаружения и прогнозы
net.setInput(blob)
detections = net.forward()

```

Первое, что мы делаем — считываем кадр из потока, затем заменяем его размер. Чуть позже нам понадобится ширина и высота, получим их сейчас. Затем следует преобразование кадра в blob с модулем dnn. И наконец мы устанавливаем blob как входные данные в нашу нейросеть и передаём эти данные через net, которая обнаруживает наши предметы.

«Фильтруем» объекты. На данный момент, мы обнаружили объекты в видео потоке. Теперь пришло время посмотреть на значения валидности и решить, должны ли мы нарисовать квадрат вокруг объекта и повесить лейбл.

```

# запускаем цикл по обнаружениям
for i in np.arange(0, detections.shape[2]):
    # получаем значение валидности
    confidence = detections[0, 0, i, 2]

    # фильтруем обнаружения если значение валидности выше заданного порога
    if confidence > args["confidence"]:
        # извлекаем индекс лейбла в классе
        # и высчитываем координаты рамки вокруг обнаруженного объекта
        idx = int(detections[0, 0, i, 1])
        box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
        (startX, startY, endX, endY) = box.astype("int")

        # рисуем цветной прямоугольник вокруг объекта
        label = "{}: {:.2f}%".format(CLASSES[idx],
            confidence * 100)
        cv2.rectangle(frame, (startX, startY), (endX, endY),
            COLORS[idx], 2)
        y = startY - 15 if startY - 15 > 15 else startY + 15
        cv2.putText(frame, label, (startX, y),
            cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)

```

Мы начинаем проходить по циклами через наши detections, помня, что несколько объектов могут быть восприняты как единое изображение. Мы также делаем проверку на валидность (т.е. вероятность) для каждого обнаружения. Если валидность достаточно велика (т.е. выше заданного порога), отображаем предсказание в терминале, а также рисуем на видео потоке предсказание (обводим объект в цветной прямоугольник и вешаем лейбл).

Оставшиеся задачи:

1. Отображение кадра
2. Проверка ключа выхода
3. Обновление счётчика FPS

```

# показать выходной кадр
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF

# если клавиша `q` была нажата, выйти из цикла
if key == ord("q"):

```

## **break**

```
# обновить FPS счетчик  
fps.update()
```

Код довольно очевиден: во-первых, выводим кадр. Затем фиксируем нажатие клавиши, проверяя, не нажата ли клавиша «q» (quit). Если условие истинно, мы выходим из цикла. Обновляем наш счётчик FPS.

Если происходит выход из цикла (нажатие клавиши «q» или конец видео потока), у нас есть вещи, которые нужно сделать:

```
# остановить таймер и отобразить информацию FPS  
fps.stop()  
print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))  
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))  
  
# сделать небольшую уборку  
cv2.destroyAllWindows()  
vs.stop()
```

При выходе из цикла, останавливаем счётчик FPS и выводим информацию о конечном значении FPS в терминал. Закрываем окно программы, прекращая видео поток.

## ЗАКЛЮЧЕНИЕ

В течении выполнения дипломной работы было проведено небольшое исследование нынешних систем обработки изображений.

Дипломная работа посвящена исследованию особенностей регистрации визуальной информации в системе обработке изображений и разработке алгоритма для распознавания объектов в реальном времени.

Рассмотрены основные понятия компьютерной обработки оптического изображения, понятия теории сигналов, дискретизация и квантование непрерывных изображений, а так же свободная библиотека компьютерного зрения OpenCV.

На основе полученных знаний в ходе исследования данной темы был разработан алгоритм распознавания объектов. Данный алгоритм может быть использован в качестве методических указаний для лабораторных работ по изучению компьютерных систем.

В завершении проделанной работы, поставленные в данной работе задачи, были успешно выполнены.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Процесс получения визуальной информации человеком и системой компьютерного зрения [Электронный ресурс]. URL: <https://intellect.icu/13-protsess-polucheniya-vizualnoj-informatsii-chelovekom-i-sistemoj-kompyuternogo-zreniya-osobnosti-registratsii-vizualnoj-informatsii-v-sisteme-obrabotki-izobrazhenij-5930>
2. Бакут, П.А. Сегментация изображений: Методы выделения границ областей / П.А. Бакут, Г.С. Колмогоров // Зарубежная электроника, 1987, № 10, С. 25-47
3. Lee J. S. Computer Vision, Graphics and Image Processing, 1983, v. 24, № 2, P. 255-269.
4. Даджион, Д. Цифровая обработка многомерных сигналов / Д. Даджион, Р. Мерсеро: Пер. с англ. под ред. Л.П. Ярославского. М.: Мир, 1988. -488 с.
5. Путятин, Е.П. Обработка изображений в робототехнике / Е.П. Путятин, С.И. Аверин. -М.: Машиностроение, 1990. 320 с.
6. OpenCV [Электронный ресурс]. URL: <https://opencv.org/>
7. Roberts L. G. In: Optical and Electrooptical Information Processing Ed. by J. Tippet, D. Berkowitz. - MIT Press, 1965, P. 159-197
8. Robinson G. S. Computer Graphics and Image Processing, 1977, v. 6, № 5, P. 492-501.
9. Гонсалес, Р. Цифровая обработка изображений в среде MATLAB / Р. Гонсалес, Р. Вудс, С. Эддинс: пер. с англ. под ред. В.В. Чепыжова. М.: Техносфера. -2006. - 616с. - ISBN 5-94836-092-X.
10. Getting started with Caffe [Электронный ресурс]. URL: <http://on-demand.gputechconf.com/gtc/2015/webinar/deep-learning-course/getting-started-with-caffe.pdf>
11. Прэтт У. Цифровая обработка изображений: В 2 кн. М.: Мир, 1982. Кн.1 312 с.
12. Прэтт У. Цифровая обработка изображений: В 2 кн. М.: Мир, 1982. Кн.2. 480 с.
13. Чэн Ш. -К. Принципы проектирования систем визуальной информации: Пер. с англ. М.: Мир, 1994. - 408 с.
14. Smathers R.L. and Brody W.R. Digital radiology: current and future trends // Br. J. Radiol. -1985. v.8. - P. 285-307.
15. Василенко, Г.И. Теория восстановления сигналов / Г.И. Василенко. -М.: Сов. радио, 1979.-272 с.
16. Методы компьютерной обработки изображений / Под. ред. В.А. Сойфера. 2-е изд., испр. - М.: ФИЗМАТЛИТ. - 2003. - 784 с. - ISBN5.9221-0270-2.

## ПРИЛОЖЕНИЕ А

```
# python real_time_object_detection.py --prototxt
MobileNetSSD_deploy.prototxt.txt --model MobileNetSSD_deploy.caffemodel

from imutils.video import VideoStream
from imutils.video import FPS
import numpy as np
import argparse
import imutils
import time
import cv2

ap = argparse.ArgumentParser()
ap.add_argument("-p", "--prototxt", required=True,
                help="path to Caffe 'deploy' prototxt file")
ap.add_argument("-m", "--model", required=True,
                help="path to Caffe pre-trained model")
ap.add_argument("-c", "--confidence", type=float, default=0.2,
                help="minimum probability to filter weak detections")
args = vars(ap.parse_args())

CLASSES = ["background", "aeroplane", "bicycle", "bird", "boat",
           "bottle", "bus", "car", "cat", "chair", "cow", "diningtable",
           "dog", "horse", "motorbike", "person", "pottedplant", "sheep",
           "sofa", "train", "tvmonitor"]
COLORS = np.random.uniform(0, 255, size=(len(CLASSES), 3))

print("[INFO] loading model...")
net = cv2.dnn.readNetFromCaffe(args["prototxt"], args["model"])

print("[INFO] starting video stream...")
vs = VideoStream(src=0).start()
time.sleep(2.0)
fps = FPS().start()

while True:

    frame = vs.read()
    frame = imutils.resize(frame, width=400)

    (h, w) = frame.shape[:2]
    blob = cv2.dnn.blobFromImage(cv2.resize(frame, (300, 300)),
                                 0.007843, (300, 300), 127.5)

    net.setInput(blob)
    detections = net.forward()

    for i in np.arange(0, detections.shape[2]):
        confidence = detections[0, 0, i, 2]

        if confidence > args["confidence"]:

            idx = int(detections[0, 0, i, 1])
            box = detections[0, 0, i, 3:7] * np.array([w, h, w, h])
            (startX, startY, endX, endY) = box.astype("int")
            label = "{}: {:.2f}%".format(CLASSES[idx],
                                       confidence * 100)
            cv2.rectangle(frame, (startX, startY), (endX, endY),
                        COLORS[idx], 2)
            y = startY - 15 if startY - 15 > 15 else startY + 15
            cv2.putText(frame, label, (startX, y),
                       cv2.FONT_HERSHEY_SIMPLEX, 0.5, COLORS[idx], 2)
```

## ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

```
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF

if key == ord("q"):
    break

fps.update()

fps.stop()
print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))

cv2.destroyAllWindows()
vs.stop()
```