

Введение в цифровую обработку изображений (ЦОИ)

Цифровая обработка изображений (ЦОИ) относится к цифровой обработке сигналов, где сигналом служит изображение и представляет собой самостоятельную область знаний, которая быстро развивается и охватывает большой спектр методов, которые имеют очень широкое применение. Методы обработки изображений играют значительную роль в научных исследованиях, промышленности, медицине, космических исследованиях и информационных системах. Примерами применения этих методов могут служить цифровая передача изображений с космических кораблей, повышение четкости изображений, создаваемых электронным микроскопом, коррекция искажений изображений, принимаемых из космоса, автоматический анализ характера местности, исследование природных ресурсов по фотоснимкам, передаваемым со спутников Земли, формирование и улучшение качества биологических и медицинских изображений, автоматическое составление карт по аэрофотоснимкам.

Повышение качества изображений достигается двумя видами обработки изображений: реставрацией изображений и их улучшением. Под реставрацией обычно понимают процедуру восстановления или оценивания элементов изображения, целью которой является коррекция искажений и наилучшая аппроксимация идеального неискаженного изображения. Для улучшения изображений используется комплекс операций, призванных улучшить восприятие изображения наблюдателем или же преобразовать его в другое изображение, более удобное для машинной обработки.

При машинной обработке улучшение изображения тесно связано с задачей извлечения информации. Например, система улучшения изображений производит подчеркивание границ исследуемого изображения путем высокочастотной фильтрации. На основе обработанного изображения выполняется операция прослеживания контура объекта, определяется его форма и размеры. В этом случае система улучшения изображения используется для того, чтобы подчеркнуть важнейшие признаки исходного изображения и облегчить задачу извлечения информации. Манипуляции с данными видимых изображений – это одно из основных применений обработки изображений. Обработка изображений может быть использована для того, чтобы получить видимое изображение цифровых данных, усиленных определенным образом для выделения некоторых аспектов данных. Такие типы обработки изображения используются, например, в медицинском оборудовании, гидролокаторах, радарах, ультразвуковом оборудовании, тепловизионном оборудовании и т.д.

Общие сведения о системе MATLAB

Название системы MatLab происходит от слов Matrix Laboratory (матричная лаборатория), это средство математического моделирования, обеспечивающее проведение исследований во всех известных областях науки и техники, позволяет использовать все современные достижения компьютерных технологий, в том числе средства визуализации и аудификации (озвучивания) данных, а также возможности обмена данными через Internet, имеет богатейшую библиотеку (более 800 функций). Для облегчения библиотека разбита на разделы, которые называются ToolBoxes. Функции общего назначения входят в ядро системы MatLab, другие включены в специализированные разделы. Полная комплектация включает около 30 разделов: математический пакет, цифровая обработка изображений, аппарат построения и анализа нейронных сетей и ряд других. Пакет ориентирован на обработку массивов данных.

Для системы MatLab характерны файлы нескольких типов:

- файлы с расширением m (m – файлы) – содержат тексты функций на языке MatLab. В виде m – файлов реализованы все библиотечные функции.
- файлы с расширением mdl (mdl - файлы) – файлы моделей пакета SIMULINK – инструмента визуального моделирования.
- файлы с расширением mat (mat - файлы) – файлы, содержащие данные в двоичном коде, доступ к которым возможен из командного окна системы MatLab.

В интерфейс системы MatLab по умолчанию входят пять панелей:

1. Command Window (Окно Команд), где проводятся все расчеты и операции
2. Launch Pad (Окно Разделов), где можно найти информацию о разделе и демонстрационные примеры для него.

3. (Рабочее пространство) – отображает текущий набор переменных, заведенных пользователем в командном окне.
 4. Current Directory (Текущий каталог)
 5. Command History (История команд) – хранит все команды, набираемые пользователями.
- Панель Launch Pad перекрывает панель Workspase, панель Command History перекрывает панель Current Directory и для выделения нужной панели на передний план нужно щелкнуть на соответствующей вкладке.

Формирование изображения.

Для формирования изображения могут использоваться следующие технические средства:

1. оптическое и телевизионное формирование;
2. радиолокационные технические средства;
3. использование проникающего излучения;
4. комбинированные (томография);
5. технические средства машинной графики.

В зависимости от используемых средств индикации сформированные изображения могут быть воспроизведены в виде:

1. одномерных динамических изображений (осциллограф);
2. двумерных статических изображений (фото);
3. двумерные динамические изображения (фильмы);
4. квазитрехмерных статических изображений (стерео фото);
5. квазитрехмерных динамических изображений (стерео фильмы);
6. машинной графики;

Существуют две основные формы представления изображений: непрерывная и дискретная

Непрерывное изображение может быть представлено как распределение уровней яркости в пространстве, то есть фиксированное в какой – либо плоскости отображение пространственного распределения собственного или отраженного излучения некоторого сюжета. Принято представлять изображение в виде непрерывной функции распределения значений интенсивности в фиксированной плоскости - $J(x, y, t, \lambda)$, где x, y – координаты в этой плоскости; t – время; λ – длина волны излучения - для описания цветного динамического изображения и - $J(x, y, \lambda)$, если функция не зависит от времени (т.е. изображение не меняется в течении времени наблюдения), – для описания цветного статического изображения. Если при анализе и обработке не имеет значение длина волны излучения (например, черно-белое изображение), то длину волны опускают.

Лекция №2

Основные характеристики изображения

Интенсивность

Важной характеристикой изображения является интенсивность. Величина интенсивности определяется значением конкретной физической величины в зависимости от природы изображения. Если является источником собственного или отраженного излучения (изображение на телевизионном экране, фотографическое изображение), то в качестве интенсивности используется яркость, если является промежуточным и предназначено для дальнейшего преобразования (например, с помощью фотоприемника в электрический ток), то для характеристики используется освещенность.

Контраст.

Поскольку энергия излучения не может быть бесконечно большой, то существует верхняя граница интенсивности J_{\max} , определяющий величину допустимой интенсивности реального изображения. Часто в изображении существует и нижний предел интенсивности $J_{\min} > 0$.

В этом случае разность $D_j = J_{\max} - J_{\min}$ определяет диапазон значений интенсивности или амплитуд данного изображения. Амплитудой называется значение интенсивности. Амплитудой J_{\min} называется уровнем черного в изображении, амплитуда J_{\max} - уровнем белого. Промежуточные значения амплитуд образуют полутона в изображении (шкалу серого). Такие изображения называют полутоновыми. Если изображение имеет два уровня – максимальное и минимальное, то его называют бинарным.

Для характеристики относительного изменения полутонов в соседних точках используют величину контраста. Пусть интенсивность изображения в точке (X_1, Y_1) составляет $J_1 = J(X_1, Y_1)$, а в точке (X_2, Y_2) - $J_2 = J(X_2, Y_2)$, найдем $J'_{\max} = \max(J_1, J_2)$ и $J'_{\min} = \min(J_1, J_2)$, тогда за контраст принимается следующая величина:

$$K = \frac{J'_{\max} - J'_{\min}}{J'_{\max}}$$

Иногда вместо контраста используется коэффициент модуляции:

$$K_M = \frac{J'_{\max} - J'_{\min}}{J'_{\max} + J'_{\min}}.$$

Часто при анализе конкретного изображения рассматривается некоторый интересующий объект на окружающем его фоне. Если средняя по площади яркость объекта задается величиной \bar{L}_o , а фона - \bar{L}_ϕ , то в качестве контраста и коэффициента модуляции принимаются значения:

$$K = (\bar{L}_o - \bar{L}_\phi) / \bar{L}_\phi$$

$$K_M = \left| (\bar{L}_o - \bar{L}_\phi) \right| / (\bar{L}_o + \bar{L}_\phi).$$

Чем выше контраст, тем заметнее объект или интересующая деталь на окружающем фоне.

Слабоконтрастные изображения вообще могут не различаться глазом. Нижняя граница контраста, при котором глаз еще может различать перепад яркостей на пределе своей чувствительности, называется пороговым контрастом $K_{\text{пор}}$. В широком диапазоне яркостей $K_{\text{пор}} = 0.02 \dots 0.05$. При очень темном или, наоборот, ярком фоне величина $K_{\text{пор}}$ увеличивается, то есть для различий деталей в изображении требуется больший перепад яркостей.

Пространственная частота

Пространственная частота характеризует частоту изменения яркости. Она определяется, как правило, количеством колебаний яркости на единицу длины. Часто значение пространственной частоты измеряется в линиях или штрихах на мм (лин/мм, штр/мм или прост мм^{-1}), показывающих сколько светлых и темных участков изображения с гармоническим распределением интенсивности укладывается на 1мм длины вдоль рассматриваемого направления. Чем выше пространственная частота в изображении, тем более мелкую структуру оно имеет. Изображение черно-белой

шахматной доски является примером изображения с высокой пространственной частотой. Изображения облаков обычно содержит низкую пространственную частоту.

Амплитудно – частотная характеристика

Любой периодический сигнал можно разложить в ряд Фурье. При этом он представляется в виде суммы гармонических функций либо комплексных экспонент с частотами, образующими арифметическую прогрессию.

Для одномерного сигнала разложение в ряд Фурье в вещественной форме без учета среднего значения интенсивности можно записать:

$$J(x) = \sum_{n=1}^{\infty} A_n \cos(n\omega_1 x + \varphi_n),$$

где A_n - амплитуды, а φ_n – фазы

В комплексной форме разложение в ряд Фурье записывается следующим образом

$$J(x) = \sum_{n=-\infty}^{\infty} \dot{C}_n e^{-jn\omega_1 x}$$

где

$$\dot{C}_k = \frac{1}{T} \int_{-T/2}^{T/2} J(x) \exp(-jk\omega_1 x) dx$$

Комплексные коэффициенты ряда связаны с амплитудами A_n и фазами φ_n следующими соотношениями:

$$A_n = 2 \left| \dot{C}_n \right|, \quad \varphi_n = \arg(\dot{C}_n).$$

Совокупность амплитуд гармоник ряда Фурье называется амплитудным спектром или амплитудно-частотной характеристикой, а совокупность их фаз – фазовым спектром или фазо-частотную характеристику.

Для спектрального анализа непериодических сигналов используют преобразование Фурье.

Формула для расчета коэффициентов комплексного ряда Фурье модифицируется следующим образом:

- 1 частота становится непрерывным параметром преобразования, то есть $k\omega_1$ в формуле заменяется на ω ;
- 2 удаляется множитель $1/T$;
- 3 результатом вычислений вместо коэффициентов ряда \dot{C}_k является функция частоты $S(\omega)$ - спектральная функция (плотность) сигнала.

В результате получим формулу прямого преобразования Фурье:

$$J(\omega) = \int_{-\infty}^{\infty} J(x) e^{-j\omega x} dx$$

Для вычисления обратного преобразования Фурье в формуле самого ряда Фурье суммирование заменяется интегрированием и, кроме того, перед интегралом появляется деление на 2π :

$$J(x) = \frac{1}{2\pi} \int_{-\infty}^{\infty} J(\omega) e^{j\omega x} d\omega$$

Если функция $J(x, y)$ описывает распределение энергии в пространстве (в плоскости изображения), то амплитудно-частотный спектр $J(\omega_x, \omega_y)$ описывает распределение энергии по пространственным частотам. Чем больше амплитуды высокочастотных гармоник, тем более мелкую структуру имеет изображение.

Амплитудно-частотный спектр функции $J(x, y)$ представляется спектральной плотностью $J(\omega_x, \omega_y)$, получаемой с помощью прямого преобразования Фурье:

$$J(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} J(x, y) e^{-j(\omega_x x + \omega_y y)} dx dy$$

Значение $J(\omega_x, \omega_y)$ удобнее выражать в полярных координатах:

$$J(\omega_x, \omega_y) = |J(\omega_x, \omega_y)| e^{-j\phi(\omega_x, \omega_y)}$$

где $|J(\omega_x, \omega_y)| = [R^2(\omega_x, \omega_y) + I^2(\omega_x, \omega_y)]^{1/2}$ - амплитудный спектр Фурье-преобразования

$$\phi(\omega_x, \omega_y) = \arctg \left[\frac{I(\omega_x, \omega_y)}{R(\omega_x, \omega_y)} \right] \quad - \text{ фазовый спектр Фурье-преобразования.}$$

Имея спектральную плотность можно получить изображения с помощью обратного преобразования Фурье:

$$J(x, y) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} J(\omega_x, \omega_y) e^{j(\omega_x x + \omega_y y)} d\omega_x d\omega_y.$$

Лекция №3

Контурсы

Под контуром (краем) понимается та часть изображения, в которой интенсивности изменяется быстро. Скорость изменения интенсивности $J(x, y)$ определяется первыми производными по координатам. В каждой точке изображения можно определить направление, в котором увеличение интенсивности происходит с наибольшей скоростью – градиент функции $J(x, y)$. Градиент является вектором, его модуль равен наибольшей для данной точки скорости увеличения интенсивности:

$$\overrightarrow{\text{grad}}(J(x, y)) = \frac{dJ}{dx} \vec{i} + \frac{dJ}{dy} \vec{j}$$

$$|\overrightarrow{\text{grad}}(J(x, y))| = \sqrt{\left(\frac{dJ}{dx}\right)^2 + \left(\frac{dJ}{dy}\right)^2}.$$

Шумы в изображении

Часто изображения содержат не только полезную информацию, но и различные помехи. Если такие помехи носят случайный характер, то они называются шумами. Зашумленное изображение называется стохастическое. Рассмотрим модели аддитивного, импульсного и мультипликативного шума.

Модель аддитивного шума используется, когда сигнал на выходе системы и на промежуточном этапе его преобразования рассматривается как сумма полезного сигнала и некоторого случайного сигнала. Так описывают действие зернистости фотопленки или флюктуационный шум в радиосистемах.

Если действие шума сказывается не на всей протяженности сигнала, а только в случайно расположенных точках, где значения сигнала заменяется случайной величиной, такой шум называется импульсным. Такой шум встречается в системах передачи изображений по радиоканалам с использованием нелинейных видов модуляции (например, частотная), в цифровых системах передачи и хранения изображения.

В мультипликативной модели шума полезный сигнал умножается на случайный сигнал. К такой модели относится шум фотоэлектронных умножителей, шум диффузности в голографических изображающих системах.

Аналоговые системы

Термин система используется, чтобы подчеркнуть высокоуровневый характер рассмотрения – система описывается своими числовыми и функциональными характеристиками без привлечения принципиальных схем.

Классификация систем

Важнейшим классификационным признаком является линейность или нелинейность системы.

Линейными называются системы, для которой выполняется признак суперпозиции: реакция на сумму сигналов равна сумме реакций на эти сигналы, поданные на вход по отдельности.

Системы могут быть стационарные и нестационарные, что определяет постоянства или непостоянства их характеристик во времени. Система называется стационарной (система с постоянными параметрами), если произвольная задержка подаваемого на вход сигнала приводит лишь к такой же задержке выходного сигнала, не меняя его формы.

Способы описания линейных систем

Связь между входными и выходными сигналами линейной цепи можно выразить в виде дифференциального уравнения (ДУ) вида

$$a_n \frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + a_{n-2} \frac{d^{n-2} y}{dt^{n-2}} + \dots + a_1 \frac{dy}{dt} + a_0 y(t) =$$

$$= b_m \frac{d^m x}{dt^m} + b_{m-1} \frac{d^{m-1} x}{dt^{m-1}} + b_{m-2} \frac{d^{m-2} x}{dt^{m-2}} + \dots + b_1 \frac{dx}{dt} + b_0 x(t) \quad (1)$$

здесь $x(t)$ – входной сигнал, $y(t)$ – выходной сигнал, a_i и b_i – постоянные коэффициенты. Таким образом, цепь описывается набором коэффициентов $\{a_i\}$ и $\{b_i\}$. Значение n называется порядком цепи.

Характеристики линейных систем

Импульсная характеристика - это реакция системы на поданную на вход дельта-функцию. Дельта-функция или функция Дирака, представляет собой бесконечно узкий импульс с бесконечной амплитудой, расположенный при нулевом значении аргумента функции. «Площадь» импульса равна единице:

$$\delta(t) = \begin{cases} 0, & t \neq 0, \\ \infty, & t = 0, \end{cases} \quad \int_{-\infty}^{\infty} \delta(t) dt = 1.$$

Дельта-функция обладает фильтрующим свойством. Оно состоит в том, что, если дельта-функция присутствует под интегралом в качестве множителя, то результат интегрирования будет равен значению остального подынтегрального выражения в той точке, где сосредоточен дельта-импульс.

$$\int_{-\infty}^{\infty} f(t) \delta(t - t_0) dt = f(t_0).$$

Любой сигнал может быть представлен в виде свертки самого себя с дельта-функцией:

$$s_{\text{вых}}(t) = \int_{-\infty}^{\infty} s_{\text{вх}}(t') \delta(t - t') dt'$$

линейная система преобразует относительно переменной t все функции, входящие в это выражение. В результате получается формула, показывающая, что выходной сигнал линейной системы **равен свертки входного сигнала и импульсной характеристики системы**.

$$s_{\text{вых}}(t) = \int_{-\infty}^{\infty} s_{\text{вх}}(t') h(t - t') dt'$$

где $h(t - t')$ - импульсная характеристика. Импульсная характеристика (импульсный отклик) для оптических систем называется функцией рассеяния точки. (ФРТ).

Рассмотрим структурную схему системы технического зрения (СТЗ) со встроенной ЭВМ (рис. 1).

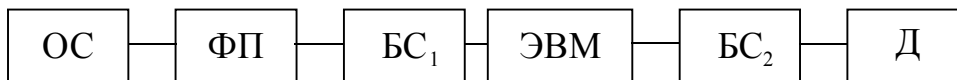


Рис. 1

Оптическая система (ОС) формирует изображение объекта в плоскости фотоприемника (ФП). Через входной блок сопряжения (БС₁) информация об изображении в виде электрических сигналов вводится в ЭВМ, где производится его обработка. Обработанная информация через выходной блок сопряжения (БС₂) поступает на вход дисплея (Д), где отображается выходное изображение. Каждое звено оптико-электронного тракта СТЗ имеет свой импульсный отклик, и изображение на экране дисплея можно рассчитать, последовательно повторяя операцию свертки импульсного отклика текущего звена с сигналом на выходе предыдущего.

Комплексный коэффициент передачи (частотная характеристика)

Для описания прохождения сигнала через линейную систему в частотной области надо выполнить преобразование Фурье свертки входного сигнала и импульсной характеристики системы. Преобразование Фурье от свертки равно произведению преобразований Фурье сворачиваемых сигналов. В результате получим следующее выражение:

$$\dot{S}_{\text{вых}}(\omega) = \dot{S}_{\text{вх}}(\omega) K(\omega)$$

где $\dot{S}_{\text{вх}}(\omega)$ и $\dot{S}_{\text{вых}}(\omega)$ - спектральная плотность входного и выходного сигнала соответственно,

$K(\omega)$ - комплексный коэффициент передачи, который является преобразованием Фурье импульсной характеристики системы:

$$\dot{K}(\omega) = \int_{-\infty}^{\infty} h(t) e^{-j\omega t} dt$$

Модуль и фаза комплексного коэффициента передачи является соответственно амплитудно-

частотной (АЧХ) и фазо-частотной (ФЧХ) характеристиками системы. Значение $K(\omega)$ показывает как изменяется при прохождении через систему комплексная амплитуда синусоиды с частотой ω .

АЧХ показывает, во сколько раз изменяется амплитуда синусоиды, а ФЧХ - каков будет полученный ею фазовый сдвиг.

Комплексный коэффициент передачи для оптических систем называется оптической передаточной функцией (ОПФ). Для получения ОПФ многозвенной системы достаточно перемножить ОПФ всех звеньев:

$$K_{\Sigma} = K_{\text{ОП}} + K_{\text{ФП}} + K_{\text{БС1}} + K_{\text{ЭВМ}} + K_{\text{БС2}} + K_{\text{Д}}.$$

Коэффициент передачи по мощности.

Коэффициент передачи по мощности равен квадрату модуля комплексного коэффициента передачи, то есть квадрату АЧХ:

$$K_M(\omega) = K(\omega) \cdot K^*(\omega) = |K(\omega)|^2.$$

Лекция №4

Функция передачи

Если применить к обеим частям дифференциального уравнения (1) преобразование Лапласа получится выражение для функции передачи :

$$H(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + b_{m-2} s^{m-2} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + a_{n-2} s^{n-2} + \dots + a_1 s + a_0}$$

здесь a_i и b_i - те же постоянные коэффициенты, что и приведенном ранее ДУ (преобразование Лапласа можно рассматривать как обобщение преобразования Фурье, при котором частота может принимать комплексные значения). Комплексный коэффициент передачи получается из функции передачи путем подстановки $s=j\omega$.

Нули и полюса функции передачи

Разложив числитель и знаменатель функции передачи на множители, мы получим функции передачи в следующем виде:

$$H(s) = k \frac{(s - z_m)(s - z_{m-1})(s - z_{m-2}) \dots (s - z_1)}{(s + p_n)(s + p_{n-1})(s + p_{n-2}) \dots (s + p_1)}$$

где $k = b_m / a_n$ - коэффициент усиления (gain), z_i - нули функции передачи (zeros), p_i - полюсы функции передачи (poles).

Квантование и дискретизация сигналов.**Аналого-цифровое и цифро-аналоговое преобразование**

Изображение непрерывно по координатам x и y , а также по амплитуде. Чтобы преобразовать эту функцию в цифровую форму, необходимо представить ее отсчетами по обеим координатам и по амплитуде. Представление координат в виде конечного множества отсчетов называется дискретизацией, а представление амплитуды значениями из конечного множества – квантованием. Обобщенная схема цифровой обработки сигналов приведена на рис. 2. На вход поступает аналоговый сигнал $s_{\text{вх}}(t)$. Его временная дискретизация и квантование по уровню производится в аналого-цифровом преобразователе (АЦП). Вообще эти два процесса – дискретизация и квантования – являются независимыми друг о друга, но они, как правило, выполняются внутри одной микросхемы. Выходным сигналом АЦП является последовательность чисел, поступающая в цифровой процессор ЦП, выполняющий требуемую обработку. Процессор выполняет различные математические операции над входными отсчетами : ранее полученные отсчеты и промежуточные результаты могут сохраняться в памяти процессора для использования в последующих вычислениях. Результатом работы процессора является новая последовательность чисел, представляющих собой отсчеты выходного сигнала. Аналоговый выходной сигнал $s_{\text{вых}}(t)$ восстанавливается по этой последовательности чисел с помощью цифро-аналогового преобразователя (ЦАП). Напряжение на выходе ЦАП имеет ступенчатую форму; при необходимости оно может быть преобразовано в плавно меняющийся выходной сигнал с помощью сглаживающего фильтра Φ .



Рис. 2

Устройства, реализуемые с помощью структуры типа рис. 2 могут иметь самый разнообразный характер.

Количество памяти для хранения цифрового изображения зависит как от разрешения так и от глубины цвета, и может быть приблизительно определено умножением количества пикселей на количество бит соотносящих одному пикселю.

Наиболее часто встречаются следующие случаи:

1. два уровня яркости (бинарное изображение)
2. 256 уровней яркости (один байт на пиксель – полутоновое изображение)
3. 65536 уровней яркости (два байта на пиксель)

4. 16,7 млн. цветов (три байта на пиксель, (красный, синий, зеленый))
5. 4,3 млн. цветов (четыре байта на пиксель, (голубой, пурпурный, желтый, черный))

Частота Найквиста

Гармонический сигнал может быть адекватно представлен дискретными отсчетами, если его частота не превышает половины частоты дискретизации $f_N = f_D / 2$.

Максимальный шаг h в пространственной решетке при отсчетах исходного изображения, должен выбираться из следующего условия:

$$h < \frac{1}{2 f_m}, \text{ где } f_m - \text{максимальная пространственная частота изображения}$$

Типы изображений

Существуют два принципиально различных подхода к описанию изображения

Изображения может быть описано математически как набор графических примитивов (круг, прямоугольник, линия и т.п.), отрисовка которых и формирует изображение. Такое изображение называется **векторным**. Если изображение описывается двумерным массивом, каждый элемент которого представляет собой некоторое описание цвета, то такое изображение называется **растровым**. Элемент растрового изображения называют пикселом (от pixel, picture element - элемент изображения) или точкой.

Существуют несколько типов растровых изображений. Они отличаются друг от друга способами представления и хранения информации о цвете или яркости пиксела в памяти. Цвет образуется в результате смешивания нескольких составляющих, которые могут быть заданы в различных цветовых системах или системах цветовых координат (color space). Термин глубина цвета (color depth) используется для обозначения того, сколько бит необходимо для хранения информации о цвете пиксела.

Объем памяти в байтах, необходимый для хранения растрового изображения, можно вычислить по формуле:

$$V = (c * r * d) / 8$$

где c- количество столбцов; r- количество строк; d- глубина цвета (бит/ пиксел).

Существуют следующие типы растровых изображений, каждый из которых предназначен для решения определенного круга задач:

- **бинарные** - (black and white) пикселы могут принимать только два значения: 0 и 1 (черный и белый цвет).
- **полутоновые** (серые или изображениями в градациях серого Intensity, Grayscale) - пикселы могут принимать одно из значений интенсивности какого-либо одного цвета в диапазоне от минимальной до максимальной интенсивности.
- **палитровые** (Indexed) - значения пикселов являются ссылками на ячейки карты цветов (colormap), которые и содержат описания цвета пиксела в некоторой цветовой системе. Подобную карту цветов часто называют палитрой (palette);
- **полноцветные** - (Truicolor, RGB) или просто цветными называют изображения, пикселы которого непосредственно хранят информацию об интенсивностях цветовых составляющих.

Пикселы изображений, представленные массивами в формате double и uint8, должны удовлетворять требованиям, приведенным в таблице 1.

таблица 1

Тип изображения	double	uint8
Бинарное	Значения 0 и 1	Значения 0 и 1
Полутоновое	Диапазон значений [0, 1]	Диапазон значений [0, 255]
Палитровое	Диапазон значений [1, размер палитры], значение 1 указывает на первую строку палитры	Диапазон значений [0, 255], значение 0 указывает на первую строку палитры
Полноцветное	Диапазон значений [0, 1]	Диапазон значений [0, 255]

Полутоновые и бинарные изображения хранятся в виде двумерных массивов. Для доступа к значению (в данном случае яркости) пиксела изображения I надо указать строку r и столбец c: I(r, c).

Полноцветные изображения хранятся в виде трехмерных массивов, где третье измерение – значения интенсивности R, G, B. Для доступа к значениям интенсивности цветовых составляющих пиксела I надо указать строку r, столбец c и номер составляющей: 1 – для R, 2 - для G, 3 - для B, например, I(r, c, 1) позволяет получить значение красной составляющей.

Палитровые изображения хранятся в виде двумерных массивов индексов. Для каждого палитрового изображения существует двумерный массив палитры. Массив палитры всегда имеет тип double и в трех его столбцах расположены интенсивности R, G, B. Пример палитрового изображения, использующего формат представления данных uint8, приведен на рис.3.

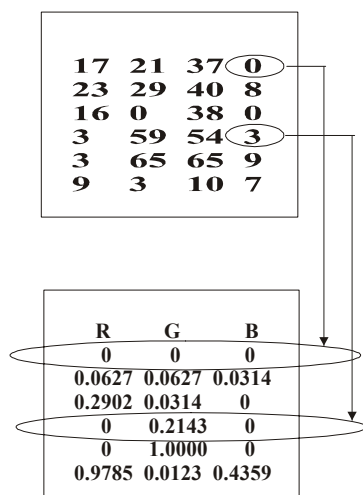


Рис. 3

Системы цветовых координат

Цветные изображения активно используются в задачах обработки изображений, поскольку они более информативны и более естественны для восприятия человеком, по сравнению с полутоновыми (черно-белыми). Рассмотрим некоторые системы цветовых координат.

Система RGB

Человеческий глаз воспринимает окружающий мир с помощью трех типов цветочувствительных элементов, называемых колбочками. Эти элементы чувствительны к излучению трех цветов: красного, синего и зеленого и все цвета, которые может видеть человек, представляются в виде их сочетания. Система координат, основанная на этих цветах, называется системой RGB. В целях стандартизации Международная комиссия по освещению присвоила следующие фиксированные значения длин волн, соответствующие координатным цветам: Red (красный) = 700 нм., Green (зеленый) = 546.1 нм., Blue (голубой) = 435.8 нм.

Система RGB реализует аддитивный процесс цветосмешения. В таком процессе конкретный цвет формируется путем смешения различных пропорций цветов, соответствующих координатным цветам, попадающим непосредственно в человеческий глаз без отражений.

Когда интенсивность всех составляющих минимальна, получается черный цвет, когда максимальна - белый. Если все составляющие имеют равную величину, получаем определенный оттенок серого цвета. Все оттенки серого расположены на линии, соединяющей точки с координатами (0,0,0) и (255,255,255). Такая система цветовых координат используется в электронно-лучевых трубках, применяемых в цветных телевизорах и компьютерных мониторах.

Система CMY и CMYK

Другой системой координат является система CMY, где C - Cyan (бирюзовый), M - Magenta (фиолетовый или пурпурный), Y - Yellow (желтый). Эта система *субтрактивная* и цвета, используемые в качестве основных цветов в этой системе координат, называются вторичными цветами. Такая система координат используется в случае, когда изображение находится на прозрачном (пленка) или непрозрачном (бумага) носителе и для рассматривания оно освещается белым светом. В таком случае, нанесение красителя определенного цвета приведет к тому, что из всего спектра падающего белого света будет вычитаться компонента, соответствующая цвету нанесенного красителя. В этом состоит принципиальное отличие аддитивной и субтрактивной систем цветовых координат.

Основными цветами красителя являются Cyan, Magenta и Yellow, тогда как вторичными цветами красителя являются Red, Green и Blue. Объекты, описанные субтрактивной системой цветовых координат, не могут быть видимы без освещения, тогда как объекты, описанные аддитивной системой цветовых координат, могут (телевизоры и компьютерные мониторы). Координаты в системе CMY могут быть получены из координат в системе RGB следующим образом:

$$\begin{vmatrix} C \\ M \\ Y \end{vmatrix} = 1 - \begin{vmatrix} R \\ G \\ B \end{vmatrix}, \quad (2)$$

Когда величина Cyan, Magenta и Yellow минимальна, т.е. нет никаких препятствий для прохождения белого света, получаем белый цвет. Когда величины Cyan, Magenta и Yellow максимальны, будут задержаны все компоненты белого света и получаем черный. Такая система координат используется в цветных принтерах.

Непосредственное применение системы CMY для принтеров имеет определенный недостаток. Он не обусловлен особенностями используемой системы координат, а поведением красителей при нанесении. Дело в том, что максимальная интенсивность всех трех компонентов должна приводить к получению чистого черного цвета. Однако из-за химического взаимодействия красителей между собой, которого невозможно полностью избежать, смесь трех компонентов обычно имеет не чистый черный цвет, а имеет какой-либо его оттенок. Для повышения качества распечатываемых изображений, было решено использовать черный цвет в виде отдельного красителя, что привело к появлению системы CMYK. Соответствующая величина четвертой компонент K – black (черный) принимается равной минимальному значению остальных

компонентов C, M и Y и вычитается из них. Например, если компонента C - минимальна, то получаем следующий переход:

$$CMY = \begin{vmatrix} C_0 & & \\ M_0 & - & C_0 \\ & Y_0 & \end{vmatrix} = \min(C_0, M_0, Y_0)$$

$$CMYK = \begin{vmatrix} C \\ M \\ Y \\ K \end{vmatrix} = \begin{vmatrix} C_0 & - & C_0 \\ M_0 & - & C_0 \\ Y_0 & - & C_0 \\ & C_0 & \end{vmatrix} = \begin{vmatrix} & 0 & \\ M_0 & - & C_0 \\ Y_0 & - & C_0 \\ & C_0 & \end{vmatrix}$$

Это приводит к тому, что общая часть, которая присутствует во всех цветах, будет представлена чистым черным цветом. Такая система координат наиболее широко распространена в современных цветных принтерах. Для дальнейшего повышения качества, к настоящему времени предложены системы и с большим - 6-8 - количеством цветов.

Система HSV

Вышеупомянутые системы цветовых координат не являются естественными, когда речь идет о восприятии изображения человеком. Несмотря на то, что глаз человека имеет конструкцию, соответствующую системе координат RGB, при оценке изображения человек использует другие параметры. Поэтому была разработана система координат, координатные оси которой соответствуют восприятию человеком цветного изображения.

В этой системе H - hue (тон), S - saturation (насыщенность), V – value (brightness, яркость).

Hue - описывает основную длину волны, описывающую цвет; *saturation* - показывает, сколько чистого белого света содержится в цвете. Чем больше чистого белого света содержится в цвете, тем меньше насыщенность. Цвета, полученные в результате спектрального разложения белого света, не содержат чистого белого света совсем и, следовательно, характеризуются максимальной насыщенности (чистые цвета). V- value - мера яркости цвета.

Система XYZ

Другая система координат, которая также была стандартизована Международной комиссией по освещению - это система XYZ.

Она может быть получена из системы RGB следующим образом:

$$\begin{cases} X = 0.4303R + 0.3416G + 0.1784B, \\ Y = 0.22197R + 0.7068G + 0.07135B, \\ Z = 0.0202R + 0.1296G + 0.9393B. \end{cases}$$

Цветовые координаты x,y,z, масштабированные так, чтобы соответствовать единичному кубу, весьма широко используются в анализе цветowych изображений из-за удобства, предоставляемого этой системой координат. Они получаются из системы XYZ следующим образом:

$$\begin{cases} x = X/(X + Y + Z), \\ y = Y/(X + Y + Z), \\ z = Z/(X + Y + Z). \end{cases}$$

Как видно из формулы, $x + y + z = 1$, следовательно, $z = 1 - x - y$. Это свойство позволяет представлять информацию о трех величинах x, y и z на одном двумерном графике (обычно в качестве осей используют x и y). Третья величина может вычисляться из первых двух.

Обработка изображений в системе MATLAB

Принятые обозначения изображения в системе MATLAB

В описании функций и примерах применяются следующие обозначения для различных типов изображений:

- **I** – полутоновые; **X** – палитровые; **RGB** – полноцветные; **BW** - бинарные
- **S** – для исходного изображения любого типа; **D** - результирующее изображение.

Символ многоточия (...) в описании функции означает, что может быть использован любой упомянутый ранее в описании набор входных или выходных параметров.

Системы координат

В пакете IPT используются две системы координат: пиксельная и пространственная.

Пиксельная система координат традиционна для ЦОИ. В ней изображение представляется матрицей дискретных пикселей. Для обращения к пикселу изображения **I** указывается номер строки **r** и номер столбца **c**, не пересечения которых расположен пиксел: **I(r,c)**. Строки нумеруются сверху вниз, а столбцы слева направо. Верхний левый пиксел имеет координаты (1 1). В пространственной системе координат изображение представляется непрерывным числовым полем, состоящим из со сторонами 1×1 . Число квадратов равно числу пикселей, координаты в центре квадрата совпадают со значением координат соответствующего пикселя в пиксельной системе координат. Для восстановления значений между дискретными пикселями используется билинейная или бикубическая интерполяция. По умолчанию координата левого верхнего угла имеет значение (0.5, 0.5). Целочисленным координатам **x** в пространственной системе координат соответствуют номера столбцов **c**, а целочисленным координатам **y** – номера строк **r** из пиксельной системы координат.

Для пространственной системы координат можно задать произвольный масштаб по осям координат с помощью двух двухкомпонентных векторов **Xdata** и **Ydata**. Пространственную систему координат с произвольным масштабом по осям удобно использовать в задачах, где требуется учесть разрешение (количество пикселей на метр) обрабатываемого изображения, т. е. представить результат в реальных метрических, а не в относительных пиксельных величинах.

Форматы представления изображений в IPT

По умолчанию в системе MATLAB есть два формата представления элементов массива:

- **double** - в виде действительных чисел двойной точности, каждый элемент формата **double** занимает 8 байт памяти.
- **uint8** - значение пикселя есть беззнаковое целое однобайтовое число, диапазон возможных значений которого [0, 255]

Представление изображения в формате **double**:

D=im2double(S) – преобразует бинарное, полутоновое или полноцветное изображение **S** в формат **double** и осуществляет приведение значений пикселей к диапазону [0,1];

X_p=im2double(X_s, 'indexed') - преобразует палитровое изображение **X_s** в формат **double**;

Представление изображения в формате **uint8**:

D=uint8(S) – преобразует бинарное, полутоновое или полноцветное изображение **S** в формат **uint8**, представляет все пиксели в виде целых неотрицательных чисел в диапазоне [0, 255].

X_p=im2uint8(X_s, 'indexed') - преобразует палитровое изображение **X_s** в формат **uint8**;

Работа с графическими форматами файлов

Чтение из файла информации об изображении

info = imfinfo(filename) - в структуре **info** возвращают информацию об изображении и способе его хранения из файла с именем **filename**.

Пакет IPT поддерживает работу со следующими графическими форматами файлов:

формат файла	Название формата
'bmp'	Windows Bitmap (BMP)
'tif или 'tiff'	Tag Image File Format (TIFF)

'jpg' или 'jpeg'	Joint Photographic Experts Group (JPEG)
------------------	---

Информация об изображении и способе его хранения в данном файле возвращается в структуре info. Структуры для разных форматов отличаются друг от друга. Общими являются первые 9 полей структуры, по которым можно определить формат файла, тип и размеры изображения:

Имя поля	Тип данных	Описание
Filename	Строка	Имя файла, если файл находится в текущей директории, или полный путь к файлу
FileModDate	Строка	Дата и время последней модификации файла
FileSize	Число	Размер файла в байтах
Format	Строка	Формат файла
FormatVersion	Строка или число	Версия формата
Width	Число	Ширина изображения в пикселах
Height	Число	Высота изображения в пикселах
BitDepth	Число	Глубина цвета изображения в битах на пиксел
ColorType	Строка	Тип изображения: 'truecolor' или 'RGB' для полноцветных изображений; 'grayscale' для полутоновых изображений; 'indexed' для палитровых изображений

В файлах форматов TIFF и HDF может храниться несколько изображений. В этом случае info является массивом структур.

Чтение изображения из файла

D = imread(filename) - читает из файла с именем filename бинарное, полутоновое или полноцветное изображение и помещает его в массив D.

[X, map] = imread(filename) - читает из файла с именем filename палитровое изображение X с палитрой map.

Прочитанное из файла изображение имеет формат представления данных uint8.

Вывод изображения на экран

imshow(I,n) – выводит на экран полутоновое изображение I, используя при выводе n уровней серого.

imshow(I,[low high]) - выводит на экран полутоновое изображение I, дополнительно **контрастируя** выводимое изображение. Пиксеты, яркость которых меньше/равна **low**, отображаются черным цветом, больше/равна **high** – белым. Все уровни серого равномерно распределены от **low** до **high**. Если вторым параметром указать пустой массив [], то **low = min(I(:)), high = max(I(:))**.

imshow(X,map) - выводит на экран палитровое изображение X с палитрой **map**.

imshow filename - выводит на экран изображение из файла с именем **filename**.

h=imshow(...) – в h будет возвращен описатель (handler) выведенного изображения (окна).

Вывод нескольких изображений в одном окне

В сочетании с функцией subplot (функция **subplot(m,n,p)** разбивает графическое окно на m×n областей, при этом m – число областей по вертикали, n - число областей по горизонтали, p – номер области, в которое будет выводиться текущий график) для вывода нескольких изображений в одно окно используется функция **subimage**:

subimage(S) – вывод в одно окно figure несколько бинарных, полутоновых или полноцветных изображений.

subimage(X,map) - вывод в одно окно figure несколько палитровых изображений X с палитрами map.

h=subimage(...) - в h будет возвращен описатель (handler) выведенного изображения (окна).

Пример 1. Вывод двух изображений S1 и S2 в одно графическое окно:

```
>> subplot(1,2,1),subimage(S1)
```

```
>> subplot(1,2,2),subimage(S2)
```

Преобразования типов изображений в IPT

Существуют ряд функций для преобразований изображений из одного типа в другой:

I=rgb2gray(RGB) – преобразование полноцветного изображения в полутоновое

newmap= rgb2gray(map) – создание палитры градаций серого из цветной

I=ind2gray(X,map) – преобразование палитрового изображения в полутоновое

[X map]=gray2ind(I,n) – преобразование полутонового изображения в палитровое, n по умолчанию равно 64

RGB= ind2rgb(X,map) - преобразование палитрового изображения в полноцветное.

[X map]= rgb2ind(I,n) – преобразование полноцветного изображения в палитровое

Для преобразования матрицы чисел в полутоновое изображение существует функция

I=mat2gray(M) – создает из матрицы чисел M полутоновое изображение I, осуществляя необходимые операции масштабирования для приведения к виду, принятому для полутоновых изображений.

Алгоритмы ЦОИ.

Манипуляции с данными видимых изображений – это одно из основных применений обработки изображений. Обработки изображений может быть использована для того, чтобы получить видимое изображение цифровых данных, усиленных определенным образом для выделения некоторых аспектов данных. Такие типы обработки изображения используются, например, в медицинском оборудовании, гидролокаторах, радарах, ультразвуковом оборудовании, тепловизионном оборудовании и т.д.

Применение алгоритма обработки к изображению не всегда делает хорошим его внешний вид. Если трансформация приносит дополнительную информацию и/или детали, невидимые в исходном изображении, результат может считаться успешным, даже если он выглядит не очень красиво.

Можно рассмотреть три класса алгоритмов обработки изображения:

- Геометрические преобразования. Они реализуются с помощью геометрических процессов. Эти процессы изменяют расположение или местонахождение элементов в изображении.
- Амплитудные преобразования. Они реализуются с помощью точечных процессов. Эти процессы изменяют значения элементов в изображении, основываясь на исходных значениях элементов и, возможно, на их позиции в изображении.
- Частотные преобразования. В основе частотных преобразований лежат пространственные процессы. Эти процессы изменяют значения элементов в изображении, основываясь на исходных значениях элементов и элементов вокруг него.

Геометрические преобразования

К геометрическим преобразованиям относятся: вырезание фрагмента, изменение размеров изображения, поворот изображения, сдвиг изображения и т.п.

Функции геометрических преобразований в системе MatLab

Кадрирование (вырезание фрагмента) изображения

D=imcrop (S)

Xd = imcrop (Xs,map)

D= imcrop (S, rect)

Xd = imcrop (Xs,map, rect)

[XData, YData, D, rect]=imcrop (...)

Функции **D=imcrop(S)** и **Xd=imcrop(Xs,map)** выводят на экран бинарное, полутоновое, полноцветное изображение **S** и политоновое изображение **Xs** и ожидают задания с помощью мыши прямоугольника, ограничивающий выражаемый фрагмент. Когда требуется задать квадратный фрагмент, следует при перемещении курсора мыши держать нажатой клавишу Shift. Если выходные параметры не заданы, например, **imcrop(I)**, то результат отображается в новом графическом окне. Если не определены входные параметры, например **D=imcrop**, то функция **imcrop** оперирует с изображением в текущем графическом окне.

Используя функции **D=imcrop(S,rect)**, **Xd= imcrop(Xs,map,rect)**, можно прямо определить ограничивающий прямоугольник, где **rect** – вектор из четырех элементов: **[Xmin Ymin w,h]**, которые задают положения верхнего левого угла (**Xmin Ymin**) прямоугольника, а также его ширину (**w**) и высоту (**h**) в пространственных координатах.

Если определить дополнительные выходные параметры, то функция **[XData, YData, D, rect]=imcrop (...)** дополнительно возвратит диапазоны изменения значений по осям координат **XData** и **Ydata** и координаты ограничивающего прямоугольника **rect**.

Изменение размеров изображения

D=imresize (S, m, method)

Функция создает изображение **D**, размеры которой в **m** раз отличаются от размеров исходного изображения **S**. Если **m** принадлежит диапазону от 0 до 1, то **D** меньше **S**. Если **m** больше 1, то **D** больше **S**.

Метод дублирования пикселей, который используется для получения увеличенного в целое число раз изображения, является частным случаем интерполяции по ближайшему соседу. Например, для увеличения изображения вдвое вначале дублируется каждый столбец, получая изображение с удвоенным горизонтальным размером. Затем дублируется каждая строка, чтобы вдвое увеличить вертикальный размер изображения. Для увеличения изображения в целое число раз применяется аналогичная процедура. Хотя увеличение с интерполяцией по ближайшему соседу выполняется быстро, оно имеет ту нежелательную особенность, что может приводить к заметной ступенчатости, особенно при большой кратности увеличения. Немного более сложным способом присвоения яркости элементам увеличенного изображения является билинейная интерполяция, в которой используются четыре ближайших соседа данной точки. Пусть (x', y') – координаты точки на увеличенном изображении. Пусть $v(x', y')$ – уровень яркости этой точки. В случае билинейной интерполяции он задается соотношением

$$v(x', y') = ax' + by' + cx'y' + d$$

где коэффициенты a, b, c, d находятся из системы четырех линейных уравнений с четырьмя неизвестными, составленными для четырех ближайших соседей точки (x', y') .

Для интерполяции может использоваться большее число соседей, что позволяет аппроксимировать дискретные точки поверхностью более сложного вида.

Для изменения размеров используется один из предопределенных методов интерполяции, который задается во входном параметре **method** в виде одной из следующих строк:

‘**nearest**’ - использовать значение ближайшего пиксела (установлено по умолчанию, и данный параметр может быть опущен при вызове функции);

‘**bilinear**’ - использовать интерполяцию по билинейной поверхности;

‘**bicubic**’ – использовать интерполяцию по бикубической поверхности.

Поворот изображения

D=imrotate(S, angle, method)

Создает изображение **D**, соответствующее повернутому исходному изображению **S**, используя один из предопределенных методов интерполяции (см. функцию **imresize**).

Угол поворота **angle** задается в градусах. Положительные значения данного параметра соответствуют повороту против часовой стрелки, а отрицательные – по часовой.

Амплитудные преобразования

Точечные процессы

Точечные процессы – это алгоритмы, которые изменяют значения элементов в изображении и основаны только на том, элемент изображения это величина. Алгоритмы точечных процессов сканируют изображение элемент за элементом, осуществляя преобразование элементов изображения.

К точечным процессам относится:

- уменьшение количества уровня серого
- соляризация
- яркостной срез
- псевдостереоскопическая имитация
- просветление
- усиление контраста
- инверсия
- локальная операция итеративного изображения

Для записи формулы алгоритма для каждого преобразования используются следующие обозначения:

\forall - для каждого элемента

B_{ij} - выходное изображение, A_{ij} – входное изображение

N – число уровней яркости

K' - максимальное число градаций

1. операторы уменьшения уровня серого

Это преобразование может привести к увеличению контраста изображения. Здесь N – требуемое количество уровней серого.

$$\forall A_{ij}, B_{ij} = \frac{A_{ij} N}{K'}$$

2. соляризация

$$\forall A_{ij} \quad B_{ij} = \frac{4A_{ij}K' - A_{ij}}{K'}$$

После применения функции соляризации участки исходного изображения с уровнем белого или близким к нему уровнем яркости имеют уровень черного, при этом сохраняются уровни черного и его участки на исходном изображении. Уровень белого на выходе приобретают участки, имеющие на входе средний уровень яркости.

3. яркостной срез (псевдораскрашивание)

Этот метод помогает выделить определенный диапазон яркости или опознать элементы изображения в определенном диапазоне для различных целей. Например, если известно, что определенные значения элементов изображения отображают различные температуры, то псевдораскрашивание можно использовать для лучшего различения температур.

$$\forall A_{ij} \quad B_{ij} = \begin{cases} 0, & A_B < A_{ij} < A_H \\ A_{ij}, & A_H \leq A_{ij} \leq A_B \end{cases} \quad \text{или} \quad B_{ij} = \begin{cases} 0, & A_B < A_{ij} < A_H \\ K, & A_H \leq A_{ij} \leq A_B \end{cases}$$

где A_H - нижнее граничное значение определяемого диапазона яркости, A_B - верхнее граничное значение определяемого диапазона яркости.

Если установлена корреляция (связь) между значением элементов изображения и информационным содержанием, то применение псевдораскрашивания должно дать хорошие результаты.

4. псевдостереоскопическая имитация

$$\forall A_{ij} \quad B_{ij} = (k' - A_{ij}) A_{i+N, j+N} / k'$$

N – выбирается из 1, 2, 3

5. Просветление

Иногда изображение может быть улучшено с помощью регулировки яркости. Просветление – это точечный процесс, который прибавляет (или вычитает) постоянные значения k (или от) элементам в изображении. Алгоритм может быть записан следующим образом:

$$\forall A_{ij} \quad B_{ij} = B_{ij} + b,$$

где b – постоянная яркости, которая может быть либо положительной, либо отрицательной. Если b положительна, то яркость элементов изображения возрастает, если b отрицательна, то яркость уменьшается. Такая регулировка яркости вызывает скольжение гистограммы вправо – при положительных b и влево – при отрицательных. Поэтому этот алгоритм еще называют скольжением гистограмм.

Изменение контраста

Слабый контраст – наиболее распространенный дефект фотографических и телевизионных изображений, обусловленный ограниченностью диапазона воспроизводимых яркостей, нередко сочетающийся с нелинейностью характеристики передачи уровней. Во многих случаях контраст можно повысить, изменяя яркость каждого элемента изображения.

Повышение контраста производит отображение входного диапазона яркостей изображения в выходной диапазон яркостей. Наиболее простым и распространенным способом является отображение входного диапазона яркостей в максимально допустимый диапазон яркостей с помощью линейной функции (рис.4).

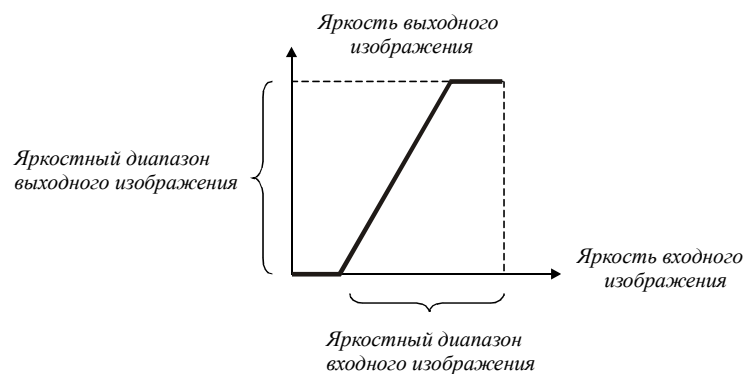


Рис. 4

В результате такой операции мы получаем большую разность в яркостях соседних пикселей, что облегчает выделение перепадов яркостей и, как правило, более комфортное зрительное восприятие.

Диапазон яркостей выходного изображения, подвергнутого цифровой обработке по изменению контраста, отличается от диапазона яркостей исходного изображения. Значения яркости в диапазоне **[low,high]** исходного изображения преобразуются в значения яркости в диапазоне **[bottom top]**. Значения яркости, меньшие low, принимают значения bottom, а значения яркости, большие high принимают значения top (рис. 5).

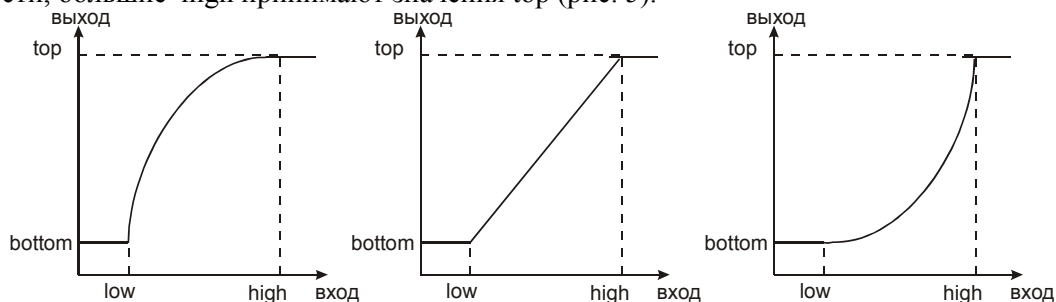


Рис. 5 - Пример контрастного линейного и нелинейного масштабирования

На рис. 6 показана характеристика передачи уровней, обеспечивающая обращение шкалы яркостей. Темные участки входного изображения будут переведены в светлые и наоборот.

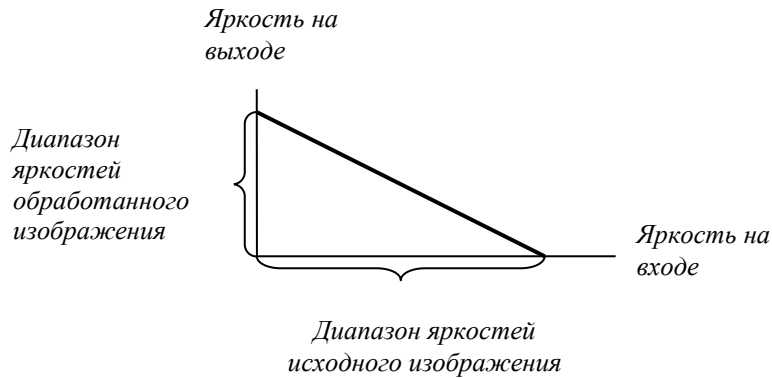


рис. 6

Алгоритм увеличения контраста может быть записан следующим образом:

$$\forall A_{ij} \quad B_{ij} = (A - C) * K,$$

где C может быть минимальным или средним значением пикселей матрицы изображения.

Алгоритм получения негативного изображения может быть записан следующим образом:

$$\forall A_{ij} \quad B_{ij} = MA - A_{i,j}$$

где MA – максимальное значение.

Для того, чтобы сделать этот алгоритм более "подвижным", требуется пороговое значение, которое показывает программе, при каком значении элементов изображения начнется негативное изображение.

Локальная операция итеративного усиления контраста изображения.

$$A_{\min}, \text{если } |A_{ij} - A_{\min}| < |A_{ij} - A_{\max}|$$

$$\forall A_{ij}, B_{ij} = A_{ij}, \text{если } |A_{ij} - A_{\min}| = |A_{ij} - A_{\max}|$$

$$A_{\max}, \text{если } |A_{ij} - A_{\min}| > |A_{ij} - A_{\max}|$$

где A_{\min} – минимальное значение интенсивности, A_{\max} – максимальное значение интенсивности.

Функции амплитудных преобразований в системе MatLab

Уменьшение количества цветов палитрового изображения

[X_D, newmap]=imapprox(X_S, map, tol,differ_option)

[X_D, newmap]=imapprox(X_S, map, n)

Функция **imapprox** создает новое палитровое изображение X_D из исходного X_S, уменьшая количество используемых цветов.

Функция **X_D, newmap]=imapprox(X_S, map, tol,differ_option)**, где **tol** – число в диапазоне [0, 1], создает палитровое X_D с палитрой newmap из палитрового X_S с палитрой map, устанавливая равномерную палитру newmap из цветов, которые в диапазоне [0, 1] берутся с шагом tol по каждой из цветовых составляющих R, G, B. Цвета исходного изображения заменяются ближайшими из установленной палитры newmap. Максимально возможное число цветов в палитре newmap составляет $(\text{floor}(1/\text{tol})+1)^3$. Действительное число цветов в палитре может оказаться меньше максимально, так как из палитры удаляются все цвета, которые отсутствуют в изображении.

Функция **[X_D, newmap]=imapprox(X_S, map, n)**, где n – число, большее 1, создает палитровое изображения X_D с палитрой newmap из палитрового X_S с палитрой map, используя для подбора палитры newmap из n цветов один из способов оптимального подбора палитры. Действительное число цветов в палитре может оказаться меньше n, так как из палитры удаляются все цвета, которые отсутствуют в изображении.

Параметр **differ_option** позволяет применять или отказываться от диффузионного псевдосмещения цветов. Может быть равен одной из следующих строковых констант:

'dither' – использовать диффузионное псевдосмещения цветов.

'nodither' – не использовать диффузионное псевдосмещения цветов.

По умолчанию применяется диффузионное псевдосмещения цветов.

Диффузионное псевдосмещения цветов создает впечатление, что на изображении присутствует большее количество различных цветов (или градаций серого цвета в случае полутоновых изображений), чем это есть на самом деле. Требуемый визуальный эффект достигается за счет группирования в локальной области пикселей с цветами, представленными в палитре, смещение которых дало бы близкий к необходимому цвет.

Для всех способов преобразования, если размер палитры map меньше либо равен 256, результирующее изображение X_D имеет формат uint8, иначе оно имеет формат double.

Построение для изображений линий уровня

imcontour(I,n)

imcontour(I,v)

imcontour(...,LineSpec)

[C,h]=imcontour(...)

Функция **imcontour(I,n)** рассматривает полутоновое изображение I как трехмерную поверхность (x, y пространственные системы координат и третья координата – яркость), строит для поверхности линии уровней и выводит их на экран. Линии уровней, соответствующие различным яркостям, обозначаются разными цветами.

Если задан параметр n (**imcontour(I,n)**), то используется n плоскостей, проходящих через значения, равномерно распределенные по оси яркости, если n опущен, то он выбирается автоматически.

Функция **imcontour(I,v)** строит линии уровней, проходящие через значения на оси яркости из вектора v.

Функция **imcontour(...,LineSpec)** дополнительно определяет тип линии в параметре **LineSpec**. Существующие типы линий можно посмотреть в функции plot.

Функция **[C,h]=imcontour(...)** возвращает массив C координат линий уровня и вектор h описателей графических объектов для изображений, соответствующих каждой линии уровня.

Пример. Построение линий уровня и получение части изображения с помощью векторов h описателей графических объектов для изображений, соответствующих каждой линии уровня.

Пример. Построение для изображений линий уровня

```
>> [X,map]=imread('c:\athena.bmp');
```

```
>> I=ind2gray(X,map);
```

```
>> figure,imshow(I)
```

```
>> figure,imcontour(I,3)
>>% get(h(i)).'<Property>' определение заданного свойства описателя графического объекта h
>> figure
>> for i=1:100
V=get(h(i),'Vertices');
F=get(h(i),'Faces');
R=get(h(i),'FaceColor');
patch('Vertices',V,'Faces',F,'FaceColor',R); % формирование и построение полигональных
% тел

hold on
end
```

Управление яркостью палитры (просветление)

brighten(beta)

newmap= brighten(beta) / newmap= brighten(map,beta)

brighten(h,beta)

Функция **brighten(beta)** заменяет текущую палитру на более яркую, если **beta** принадлежит диапазону $[0, 1]$ и более темную, если **beta** принадлежит диапазону $[-1, 0]$. Меняется только яркость цветов палитры, значение цветового фона остается неизменной.

Функции **newmap=brighten(beta)** и **newmap=brighten(map,beta)** возвращают палитру, полученную преобразованием текущей палитры или палитры **map**, но не меняют текущую палитру.

Функция **brighten(h,beta)** изменяет яркость палитры, используемой для окна с описателем **h**.

Пример по управление яркостью палитры

```
>> map1=brighten(0.5);
>>subplot(1,3,1),subimage(D,map);
>>subplot(1,3,2),subimage(D,map1);
```

Контрастирование с гамма-коррекцией

I_D= imadjust(I_s,[low,high],[bottom top],gamma)

Функция **I_D= imadjust(I_s,[low,high],[bottom top],gamma)** создает полутоновое изображение **I_D** путем контрастирования исходного полутонового изображения **I_s**. Значения яркости в диапазоне **[low,high]** преобразуются в значения яркости в диапазоне **[bottom top]**. Значения яркости, меньшие **low**, принимают значения **bottom**, а значения яркости, большие **high** принимают значения **top**. Значения **top**, **bottom**, **low**, **high** должны принадлежать диапазону $[0,1]$. Если в качестве второго или третьего параметров передать пустой вектор **[]**, то по умолчанию будет использоваться вектор $[0,1]$. С помощью показателя **gamma** осуществляется преобразование, называемое гамма-коррекцией. Параметр **gamma** определяет форму кривой характеристики передачи уровней яркости. Если **gamma** меньше 1 (Рис. 7), то характеристика передачи уровней будет выпуклой и результирующее изображение будет светлее, чем исходное. Если **gamma** больше 1, то характеристика передачи уровней будет вогнутой и результирующее изображение будет темнее, чем исходное. По умолчанию **gamma** равен 1, что соответствует линейной характеристике передачи уровней и отсутствию гамма-коррекции.

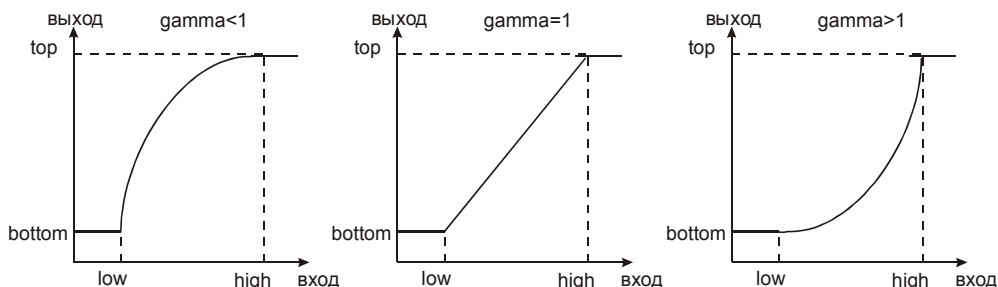


Рис. 7

Пример. Контрастирование с гамма-коррекцией

```
>> I0=imadjust(I,[0 0.9],[],1); % контрастирование полутонового изображения
>> I2=imadjust(I,[0 0.9],[],0.5); % с разными коэффициентами
>> I3=imadjust(I,[0 0.9],[],2); % гамма-коррекции
```


Лекция №11

Гистограммы

Гистограмма – это диаграмма распределения интенсивности элементов в изображении. Гистограмма может свидетельствовать об общей яркости и контрасте изображений, поэтому являются ценным методом как количественной, так и качественной обработки изображений.

Гистограмма цифрового изображения – это дискретная функция, описывающая частоту появления (вероятность) уровня серого в изображении, представленная в виде графика. По оси абсцисс откладываются номера градаций уровней серого по возрастанию (значения интенсивности), а по оси ординат – количество пикселей, имеющих данный уровень серого (частоту появления данной интенсивности). Гистограммы не попадают ни в одну из четырех алгоритмов обработки изображения, но они являются очень полезным методом для использования и контроля за результатами точечных процессов.

Гистограммы являются идеальным методом для проверки контраста изображения. Изображения могут быть сгруппированы в одну из трех категорий контраста: изображения с низким, с хорошим и с высоким контрастом.

Изображения с низким контрастом: состоят из тонов ограниченного диапазона (оттенки серого) и обычно либо слишком ярки, либо слишком темны (похожи на переэкспонированные или недоэкспонированные фотоснимки). На гистограммах это условие регистрируется, когда все образцы изображения слегка группируются друг с другом и занимают только малую часть возможных значений элементов изображения образца. Если группировка значений элементов изображения направлена в левую сторону гистограммы (по направлению к низким значениям образца), изображение будет черным, если в правую – ярким, если в центре – оно не будет ни слишком темным, ни слишком светлым.

Изображения с хорошим контрастом имеет широкий динамический диапазон. Гистограммы показывают относительно однородное распределение значений элементов изображения без больших пиков или впадин.

Как и изображения с хорошим контрастом, изображения с высоким контрастом содержит широкий диапазон серых оттенков. Однако изображения с высоким контрастом имеют большие площади, на которых преобладает темный цвет, и большие площади, которые имеют светлую окраску.

ГЭГ – глобальная эквализация гистограммы

Гистограмма распределения яркости типичного изображения имеет ярко выраженный перекося в сторону малых уровней. Яркость большинства элементов ниже среднего, на темных участках детали часто оказываются неразличимыми. Одним из методов улучшения является изменение гистограммы. Это преобразование, направленное на выравнивание гистограммы яркости называется эквализацией (ГЭГ – глобальная эквализация гистограммы). Оно может быть представлено следующим выражением:

$$\forall A_{ij} \quad B_{ij} = \frac{k'}{m \times n} \left(\sum_{L=0}^{A_{ij}-1} H(L) + \frac{H(A_{ij})}{2} \right)$$

где $H(A_{ij})$ – число элементов, имеющих яркость (интенсивность) A_{ij} ,

k' - максимальное число уровней

L – массив со значениями частоты появления яркостей (интенсивностей) в изображении.

ЛЭГ - локальная эквализация гистограммы

Преобразование ГЭГ оказывается малоэффективным, если изображение характеризуется зависимостью его качества от координат изображения. В этом случае целесообразно применять локальные (адаптивные) преобразования, т. е. преобразования, применяемое к центральному элементу D_{ij} группы элементов изображения, называемой областью примыкания (двумерная матрица значений элементов). Для выполнения ЛЭГ в каждой преобразуемой точке D_{ij} изображения на области примыкания $n \times m$ формируется локальная гистограмма $H_{i,j}(L)$ распределения уровней серого, по которой выполняется локальное преобразование:

$$\forall D_{ij} \quad B_{ij} = \frac{K'}{n \times m} \sum_{l=0}^{D_{ij}} H_{ij}(L)$$

Функции построения и выравнивания гистограммы (эквализация)

Построение гистограммы

imhist(I,n)

imhist(X, map)

[h, cx]=imhist(I, n)

[h,cx]=imhist(X, map)

Функция **imhist(I, n)** в текущем окне строит гистограмму яркостей пикселей соответственно полутонового изображения. Гистограмма состоит из **n** столбцов. По умолчанию **n=256** для полутонового изображения и **n=2** для бинарного изображения. Под рисунком гистограммы выводится шкала яркостей.

Функция **imhist(X, map)** в текущем окне строит гистограмму индексов пикселей палитрового изображения **X**. Под рисунком гистограммы выводится палитра **map**.

Функции **[h, cx]=imhist(...)** возвращают вектор гистограммы **h** и вектор положения центров столбцов гистограммы **cx** на оси яркостей (для полутоновых и бинарных изображений) или на оси индексов (для палитровых изображений), что позволяет производить дальнейшую обработку гистограммы **h** или, например, построить огибающую гистограммы с помощью функции **plot(cx, h)**.

Построение гистограммы для изображения из предыдущего примера:

```
>>imhist(D, map); % для палитрового изображения
```

Пример

```
>> [h,cx]=imhist(I);
```

```
>> figure,plot(cx,h);
```

Улучшение изображения

Эквализация изображения (выравнивание гистограммы)

ID= histeq(Is, hgram)

ID=histeq(Is, n)

[ID,T]=histeq(Is, ...)

newmap=histeq(X, map, hgram)

newmap=histeq(X, map)

Функция **histeq** улучшает контраст изображения с помощью преобразования значения пикселей исходного изображения таким образом, чтобы гистограмма яркостей пикселей результирующего изображения приблизительно соответствовала некоторой предопределенной гистограмме **hgram**. Количество элементов в **hgram** задает число столбцов гистограммы, а значение каждого элемента – относительную высоту соответствующего столбца. Значение элементов вектора **hgram** должны быть в диапазоне [0, 1]. Данная функция предназначена для преобразования **полутоновых** изображений или палитр **палитровых** изображений. Гистограмма результирующего изображения **ID**, будет лучше соответствовать заданной гистограмме **hgram** в том случае, когда количество столбцов **hgram** (**length(hgram)**) много меньше количество градаций яркости исходного изображения **Is**.

Функция **ID=histeq(Is, n)** преобразует исходное полутоновое изображение **Is** таким образом, что результирующее полутоновое изображение **ID** имело гистограмму яркости пикселей, близкую к равномерной. Чем меньше **n** по сравнению с количеством градаций яркости в изображении **Is**, тем более равномерной получается гистограмма яркостей пикселей результирующего изображение **ID**. По умолчанию значение **n** равно 64, и данный параметр можно не указывать при вызове функции.

[ID,T]=histeq(Is, ...) дополнительно возвращает вектор **T**, задающий характеристику передачи уровней яркости исходного изображения **Is** в уровни яркости результирующего изображения **ID**.

Функция **newmap=histeq(X, map, hgram)** преобразует палитру **map**, связанную с палитровым изображением **X**, таким образом, чтобы гистограмма яркостей пикселей изображения **X** с палитрой **newmap** приблизительно соответствовала гистограмме **hgram**. Функция **newmap=histeq(X, map)** преобразует палитру **map**, связанную с палитровым изображением **X**, таким образом, что гистограмма яркостей пикселей изображения **X** с палитрой **newmap** была близка к равномерной.

Пример демонстрирует повышение контраста изображения с помощью выравнивания гистограммы.

Чтения исходного изображения и вывод его на экран.

```
>> [D,map]=imread('c:\athena.bmp');  
>> newmap=histeq(D,map); % выравнивание гистограммы  
>> subplot(1,2,1),subimage(D,map);  
>> subplot(1,2,2),subimage(D,newpmap);
```

Частотные преобразования

В основе частотных преобразований лежат пространственные процессы, которые еще называют групповыми процессами или локальными операциями. Группа элементов изображения, используемых в пространственных процессах, называется областью примыкания. Область примыкания – двумерная матрица значений элементов, каждое измерение которого имеет нечетное значение элементов (чаще всего 3×3). Преобразуемые точки расположены в центре области примыкания. Применение пространственных процессов приводит к изменению спектрального состава изображения, так как в результате этого применения удаляются или усиливаются компоненты определенной частоты, обнаруженные в изображении, что позволяет их отнести к категории пространственных фильтров. Пространственная фильтрация находит широкое применение в обработке изображений. Например, ее можно использовать для выделения некоторых признаков (выделение границ и их подчеркивание), для увеличения резкости изображения, сглаживания, размытия изображения и снятия случайных помех в изображении. Для выполнения пространственной фильтрации используют метод двумерной пространственной свертки локальной окрестности обрабатываемого элемента с линейным оператором, который называется матрицей или маской весовых коэффициентов фильтра. Операция двумерной пространственной свертки выполняется следующим образом. Маской весовых коэффициентов сканируют исходное изображение. На каждом шаге сканирования находится сумма произведений элементов маски и соответствующих элементов исходного изображения. Результат присваивается одному элементу выходного изображения. На следующем шаге сканирования находится новое значение суммы, и оно присваивается следующему элементу выходного изображения.

Сложности при расчете возникают на границах изображения, так как маска фильтра выходит краем за границы изображения. Для решения этой проблемы можно использовать несколько методов. Наиболее исчерпывающими решениями являются: 1) данные на краях изображения не учитываются или 2) данные изображения могут копироваться, для того чтобы 'синтезировать' дополнительные данные границы.

Вторая трудность связана с динамичностью диапазона новых значений преобразуемых элементов.

Если все значения элементов маски положительные, то, как правило, результат шага сканирования делится на сумму элементов маски и затем присваивается соответствующему элементу выходного изображения. Если не все значения элементов маски положительные, то для получения результата шага сканирования надо сравнить полученную сумму с минимальным и максимальным значениями элементов исходного изображения. Результат, присваиваемый элементу выходного изображения будет, равен минимальному, максимальному или значению суммы, подсчитанной на текущем шаге сканирования, в зависимости от результатов сравнения.

Низкочастотные пространственные фильтры Подавление шумов

Изображение может повреждаться шумами и помехами различного происхождения, например шумом видеодатчика, шумом зернистости фотоматериалов и ошибками в канале передачи. Их влияние можно минимизировать, пользуясь классическими методами статистической фильтрации. Другой возможный подход основан на использовании эвристических методов пространственной обработки.

Шумы видеодатчиков или ошибки в канале передачи обычно проявляются на изображении как разрозненные изменения изолированных элементов, не обладающие пространственной корреляцией. Искаженные элементы часто весьма заметно отличаются от соседних элементов. Это наблюдение послужило основой для многих алгоритмов, обеспечивающих подавление шума.

Поскольку шум пространственно декоррелирован, в его спектре, как правило, содержатся более высокие пространственные частоты, чем в спектре обычного изображения. Следовательно, простая низкочастотная пространственная фильтрация может служить эффективным средством сглаживания шумов. Массив Q размера $M \times M$ выходного изображения формируется путем дискретной свертки массива F размера $N \times N$ исходного изображения со сглаживающим массивом H размера $L \times L$ согласно формуле

$$Q(m_1, m_2) = \sum_{n_1} \sum_{n_2} F(n_1, n_2) H(m_1 - n_1 + 1, m_2 - n_2 + 1) \quad (1)$$

Такие фильтры используются как для понижения визуального шума, так и для удаления высокочастотных компонент из изображения. с тем, чтобы тщательнее исследовать содержание

низкочастотных компонент – могут быть идентифицированы более слабые изменения низкой частоты. Частота отсечки низкочастотных фильтров определяется размером и коэффициентами маски фильтра. Сумма значений коэффициентов маски низкочастотного фильтра равна 1. Этот факт важен для понимания работы низкочастотных фильтров. При рассмотрении изображения, не содержащего высокочастотную составляющую, все значения элементов постоянные или изменяются очень медленно. По мере того как маска низкочастотного фильтра проходит (сканирует) через область изображения, новое значение преобразуемых элементов вычисляется как сумма коэффициентов маски, умноженных на значения элементов изображения в области примыкания. Если все значения элементов изображения в области примыкания одинаковы (постоянны), новые значения в (отфильтрованном) изображении будут такими же, как и старые (поэтому сумма коэффициентов маски равна 1), сохраняется содержание низкочастотных компонент. По мере того как маска движется по области изображения с содержанием высокочастотных компонент, любые быстрые изменения интенсивности усредняются с остальными элементами в области примыкания, тем самым, понижая содержание высокочастотных компонент. Визуальным результатом является слабая нерезкость изображения. Низкочастотная фильтрация может использоваться для улучшения резкости изображения. Для этого отфильтрованное изображение с помощью низкочастотного фильтра вычитается из исходного изображения. В результате чего будет относительное увеличение содержания высокочастотных компонент без увеличения шума в изображении.

. Ниже приведены маски низкочастотных фильтров.

$$\begin{array}{ccc}
 \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\
 \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\
 \frac{1}{9} & \frac{1}{9} & \frac{1}{9}
 \end{array}
 \quad
 \begin{array}{ccc}
 \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \\
 \frac{1}{10} & \frac{1}{5} & \frac{1}{10} \\
 \frac{1}{10} & \frac{1}{10} & \frac{1}{10}
 \end{array}
 \quad
 \begin{array}{ccc}
 \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\
 \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\
 \frac{1}{16} & \frac{1}{8} & \frac{1}{16}
 \end{array}$$

$$\sum = 1$$

Высокочастотные пространственные фильтры

Высокочастотные фильтры выделяют высокочастотные компоненты изображения, оставляя содержание низкочастотных компонент нетронутым. Области с высокой частотой станут ярче, а части с низкой частотой станут черными. Иногда резкость изображения увеличивается за счет выделения шума изображения. При использовании высокочастотной фильтрации возможно выделение границ контуров.

Большое значение коэффициента центра маски определяет действие высокочастотного фильтра. По мере перемещения центра с большим значением коэффициента по области с высокой пространственной частотой, новое значение рассматриваемых элементов изображения многократно увеличиваются. Меньшие отрицательные значения коэффициентов, сгруппированные возле центра, уменьшают эффект большого весового коэффициента. В итоге, большие изменения интенсивности элементов изображения усиливаются, а области постоянной интенсивности элементов останутся без изменения.

Ниже приведены маски высокочастотных фильтров.

$$\begin{array}{ccc}
 -1 & -1 & -1 \\
 -1 & 9 & -1 \\
 -1 & -1 & -1
 \end{array}
 \quad
 \begin{array}{ccc}
 0 & -1 & 0 \\
 -1 & 5 & -1 \\
 0 & -1 & 0
 \end{array}
 \quad
 \begin{array}{ccc}
 1 & -2 & 1 \\
 -2 & 5 & -2 \\
 1 & -2 & 1
 \end{array}$$

$$\sum = 1$$

Выделение контуров (усиление краев)

Другим пространственным процессом с использованием свертки является выделение контуров (усиление краев) изображения. По определению край представляет собой область с большим значением интенсивности. Выделение контуров изображения используется как предварительный шаг в процессе извлечения признаков. Алгоритмы усиления краев сжимают изображение до краев, поэтому фон теряется. Яркость краев после усиления пропорциональна изменению яркости, окружающей край в исходном изображении.

Хотя усиление края в основном используется в машинном зрении, оно имеет и другие применения. Например, его можно использовать как художественный метод для изготовления оригинальных изображений, которые могут затем ретушироваться в графических редакторах для создания высокохудожественного изображения. Надо отметить, что все алгоритмы усиления края, которые используют свертку, являются линейными.

Выделение контура по Лапласу

Метод выделения контура по Лапласу отличается от других методов тем, что он не зависит от направления, т.е. контуры высвечиваются независимо от направления, дают более резкий контур, чем при других алгоритмах выделения. Дополнительно он высвечивает контуры, имеющие как положительные, так и отрицательные изменения яркости. Функция Лапласа записывается следующим образом:

$$L(f(x,y)) = d^2f/dx^2 + d^2f/dy^2, \text{ где } d^2f/dx^2 \text{ и } d^2f/dy^2 - \text{ вторые частные производные по } x \text{ и } y.$$

Для дискретных функций вторые производные могут быть аппроксимированы следующим образом:

$$d^2f/dx^2 = f(x+1) - 2 \bullet f(x) + f(x-1)$$

$$d^2f/dy^2 = f(y+1) - 2 \bullet f(y) + f(y-1).$$

Таким образом, лапласиан можно записать в следующем виде:

$$L(f(x,y)) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4 \bullet f(x,y).$$

Маски лапласиановских фильтров могут иметь следующий вид:

$$\begin{array}{ccc} 0 & -1 & 0 \\ -1 & -4 & -1 \\ 0 & -1 & 0 \end{array} \quad \begin{array}{ccc} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{array} \quad \begin{array}{ccc} -1 & -2 & -1 \\ -2 & 4 & -2 \\ -1 & -2 & -1 \end{array}$$

$$\Sigma = 0$$

Все операции по усилению края ослабляют низкочастотные пространственные компоненты изображения. Области постоянной интенсивности или линейно возрастающей становятся черными, а области быстро изменяющихся значений интенсивности ярко высвечиваются. Такие фильтры, ослабляющие низкочастотные компоненты, имеют коэффициенты, сумма которых равна нулю.

Если к центральным элементам фильтров Лапласа добавить единицу, то после фильтрации будет иметь место эффект прямого добавления свернутого изображения к исходному.

Выделение контуров операторами Превитт и Собела

Маски фильтров Превитт и Собела имеют следующий вид:

для усиления края по вертикали:

$$\begin{array}{ccc} 1 & 0 & -1 \\ c & 0 & -c \\ 1 & 0 & -1 \end{array}$$

для усиления края по горизонтали:

$$\begin{array}{ccc} 1 & c & 1 \\ 0 & 0 & 0 \\ -1 & -c & -1 \end{array}$$

$c=1$ для маски фильтра Превитт

$c=2$ для маски фильтра Собела

По результатам испытаний были сделаны выводы, что маска фильтра Собела дает ложные контуры по перепаду 90° .

Усиление края методом сдвига и разности

Этот алгоритм усиливает края изображения, сдвигая изображение на элемент, а затем вычитая сдвинутое изображение из исходного. Результатом вычитания является измерение наклона распределения яркости. В области постоянной интенсивности результатом вычитания будет наклон, равный нулю. В области с большими изменениями интенсивности, например, на крае, вычитание даст большое значение наклона, что приводит к появлению светлого элемента изображения. При этом применяют абсолютное значение функции, чтобы алгоритм сдвига и разности мог регистрировать края перехода от черного к белому, так и от белого к черному.

Для усиления изображение смещается влево на один элемент и затем вычитается из исходного. Для усиления смещается вверх на один элемент и вычитается. Для усиления вертикальных и горизонтальных краев перед вычитанием смещается влево на один элемент, а затем вверх. Хотя это приближение кажется простым, его осуществление достаточно сложно. Поэтому вместо фактического сдвига изображения используется свертка для получения того же эффекта. Ниже приведены маски:

0 0 0	0 -1 0	-1 0 0
-1 1 0	0 1 0	0 1 0
0 0 0	0 0 0	0 0 0
вертикальные края	горизонтальные края	вертикальные и горизонтальные края

Методом направленного градиента

Иногда надо высветить и другие контуры (края) изображения, не только вертикальные или горизонтальные. Могут быть важны диагональные края. Избирательное высвечивание краев в различных направлениях используется иногда для того, чтобы получить общее представление об изображении. Для высвечивания краев в восьми различных направлениях используются 8 различных масок. Эти направления называются как стороны света. Маски приведены ниже:

север	северо-восток	восток	юго-восток
1 1 1	1 1 1	-1 1 1	-1 -1 1
1 -2 1	-1 -2 1	-1 -2 1	-1 -2 1
-1 -1 -1	-1 -1 1	-1 1 1	1 1 1
юг	юго-запад	запад	северо-запад
-1 -1 -1	1 -1 -1	1 1 -1	1 1 1
1 -2 1	1 -2 -1	1 -2 -1	1 -2 -1
1 1 1	1 1 1	1 1 -1	1 -1 -1

чем больше наклон, тем ярче элемент.

Размытие изображения

Размытие изображения можно получить, используя свертку изображения с фильтром, маска с весовыми коэффициентами которого приведена ниже:

1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1

Такое усреднение вызывает уменьшение деталей изображения за счет усреднения всех значений элементов в области примыкания.

Лекция №14

Медианная фильтрация

Медианная фильтрация – пространственный процесс, не попадающий в чистом виде под категорию свертки. Она располагает элементы в области примыкания в возрастающем порядке и отбирает медиану (среднее значение) в качестве новых значений результирующего изображения. Например, медианой совокупности девяти элементов является пятый элемент этой последовательности значений яркости, упорядоченных возрастанию.

Медианная фильтрация используется для:

1. для подавления помех,
2. выделения контуров (усиления краев).

Метод усреднения с порогом для подавления импульсного шума

Метод усреднения с порогом также относится к пространственным процессам, т.е. происходит обработка центрального элемента локальной окрестности (области примыкания) по определенному алгоритму, которая поочередно повторяется для всех элементов исходного изображения. Алгоритм состоит в том, что очередному пикселу выходного изображения присваивается среднее значение яркости всех пикселов в пределах области примыкания в том случае, если разница между исходным значением центрального пиксела и средним больше заданного порогового значения, в противном случае присваивается исходное значение центрального пиксела.

Метод выделения контуров, базирующийся на пространственной производной

$$\nabla x = \frac{\partial A}{\partial x} \cos \alpha + \frac{\partial A}{\partial y} \sin \alpha$$

$$\nabla = \left[\left(\frac{\partial A}{\partial x} \right)^2 + \left(\frac{\partial A}{\partial y} \right)^2 \right]^{\frac{1}{2}} \quad - \text{пространственный градиент}$$

угол наклона контура:

$$\alpha = \arctg \left[\frac{dA}{dy} / \frac{dA}{dx} \right]$$

Для дискретной версии изображения пространственный градиент ∇ можно аппроксимировать маской 3×3 :

$$I_6 I_7 I_8$$

$$I_5 I_0 I_1$$

$$I_4 I_3 I_2$$

На основе этой маски величину пространственного градиента и угла наклона элемента контура находят:

$$\nabla = \left[(I_1 - I_5)^2 + (I_7 - I_3)^2 \right]^{\frac{1}{2}}$$

$$\alpha = \arctg \left(\frac{I_7 - I_3}{I_1 - I_5} \right)$$

При упрощении:

$$\nabla = |(I_1 - I_5)| + |(I_7 - I_3)|$$

Метод, учитывающий более удаленные соседние элементы приведен ниже:
Промежуточные значения пространственного градиента вычисляются:

$$\Delta_1 = \left[(I_9 - I_0)^2 + (I_8 - I_2)^2 \right]^{\frac{1}{2}}$$

$$\Delta_3 = \left[(I_2 - I_1)^2 + (I_0 - I_{10})^2 \right]^{\frac{1}{2}}$$

$$\Delta_5 = \left[(I_0 - I_{11})^2 + (I_6 - I_4)^2 \right]^{\frac{1}{2}}$$

$$\Delta_7 = \left[(I_8 - I_6)^2 + (I_{12} - I_0)^2 \right]^{\frac{1}{2}}$$

по маске

$$\begin{array}{ccccc} & & I_{12} & & \\ & & I_8 & I_7 & I_6 \\ I_1 & I_5 & I_0 & I_{11} & I_9 \\ & I_4 & I_3 & I_2 & \\ & & I_{10} & & \end{array}$$

На основе промежуточных значений находятся значения пространственного градиента:

$$\nabla_2 = \sqrt{(\Delta_1 - \Delta_5)^2 + (\Delta_7 - \Delta_3)^2}$$

Далее вводится понятие наклона элементов:

$$\alpha_1 = \left| \frac{I_8 - I_2}{I_9 - I_0} \right| \quad \alpha_3 = \left| \frac{I_0 - I_{10}}{I_2 - I_4} \right|$$

$$\alpha_5 = \left| \frac{I_6 - I_4}{I_0 - I_{11}} \right| \quad \alpha_7 = \left| \frac{I_{12} - I_0}{I_8 - I_6} \right|$$

Затем вычисляется пространственная производная

$$\nabla_L = \left\{ \left[\min |\alpha_1 - \alpha_5|, |\alpha_1 + \alpha_5| \right]^2 + \left[\max |\alpha_7 - \alpha_3|^2 \right], |\alpha_7 + \alpha_3|^2 \right\}^{\frac{1}{2}}$$

Результатом воздействия пространственной производной ∇_L на изображение являются белые контуры на черном фоне. Совмещенное применение операторов ∇_L и ∇_2 дает возможность улучшения изображения.

Функции системы MatLab по пространственной фильтрация изображения

Двумерная свертка

D=conv2(S,h,shape);

Вычисляет свертку изображения **S** с маской **h**. Параметр **shape** определяет размер выходного изображения и может принимать те же значения, что и аналогичный параметр функции **filter2**.

Пример демонстрирует чтение изображения и его фильтрацию с целью увеличения резкости.

```
>> [X,map]=imread('c:\technlgy.bmp');
>> I=ind2gray(X,map);
>> figure,imshow(I)
>> I=im2double(I);
>> h=fspecial('unsharp',0.5);
>> I1=conv2(I,h);
>> I2=im2uint8(I1);
>> figure,imshow(I2)
```

Двумерная линейная фильтрация

D=filter2(h,S)

Выполняет фильтрацию изображения **S** двумерным фильтром, коэффициенты которого сведены в матрицу **h**. Как правило, **S** и **D** являются полутоновыми изображениями.

За шумление изображений

Часто изображения содержат не только полезную информацию, но и различные помехи. Если такие помехи носят случайный характер, то они называются шумами. Рассмотрим модели аддитивного, импульсного и мультипликативного шума.

Модель аддитивного шума используется, когда сигнал на выходе системы и на промежуточном этапе его преобразования рассматривается как сумма полезного сигнала и некоторого случайного сигнала. Так описывают действие зернистости фотопленки или флюктуационный шум в радиосистемах.

Если действие шума сказывается не на всей протяженности сигнала, а только в случайно расположенных точках, где значения сигнала заменяется случайной величиной, такой шум называется импульсным. Такой шум встречается в системах передачи изображений по радиоканалам с использованием нелинейных видов модуляции (например, частотная), в цифровых системах передачи и хранения изображения.

В мультипликативной модели шума полезный сигнал умножается на случайный сигнал. К такой модели относится шум фотоэлектронных умножителей, шум диффузности в голографических изображающих системах.

Добавление шума

ID=imnoise(Is, type, params)

Функция **imnoise** создает новое изображение **ID** путем добавления шума к исходному полутоновому изображению **Is**. В основном данная функция предназначена для создания тестовых изображений, используемых при выборе методов фильтрации шума, однако, может применяться и для придания фотореалистичности искусственно синтезированным изображениям.

Функция может добавлять шум трех типов. Тип шума задается строковой константой **type**, которая может принимать одно из следующих значений:

‘gaussian’ - для гауссова белого шума;

‘salt & pepper’- для импульсного шума (шум типа “соль и перец”);

‘speckle’- для мультипликативного шума.

Количество и смысл параметров **params** определяются выбранным типом шума. Если параметры не указаны, то будут использованы значения по умолчанию.

ID=imnoise(Is, ‘gaussian’, m,v) добавляет к изображению **Is** гауссовый белый шум с математическим ожиданием **m** и дисперсией **v**. По умолчанию математическое ожидание равно 0, а дисперсия-0.01.

ID=imnoise(Is, ‘salt & pepper’, d) добавляет к изображению **Is** импульсный шум. Параметр **d** определяет плотность шума и равен доле искаженных пикселей. По умолчанию **d=0.05**.

ID=imnoise(Is, 'spekle',v) добавляет к изображению **Is** мультипликативный шум $ID=Is+n*Is$, где n – равномерно распределенный случайный шум с математическим ожиданием 0 и дисперсией v . По умолчанию $v=0.04$.

Медианная фильтрация

I_D=medfilt2(Is,[m,n])

X_D= medfilt2(X_S, 'indexed',...)

Каждый пиксел результирующего полутонового изображения **I_D** формируется следующим образом: пикселы исходного полутонового изображения **I_S**, соответствующие всем элементам маски фильтра размера $m \times n$, составляют упорядоченную последовательность **A**. Пикселу **I_D (r,c)**, где **r** и **c** – координаты текущего положения центрального элемента маски, присваивается значение медианы последовательности **A**. Операция применяется для всех положений маски. Обработка палитровых изображений производится аналогично.

Задание маски предопределенного фильтра

h= fspecial (type, P1, P2)

Функция $h = \text{fspecial}(\text{type}, P1, P2)$ возвращает маску **h** предопределенного двумерного линейного фильтра, задаваемого строкой **type**. В зависимости от типа фильтра для него могут быть определены один или два дополнительных параметра **P1, P2**. Может быть задан размер маски **n** (если **n** - вектор, то размер маски $n(1) \times n(2)$, если **n** - скаляр, то размер маски - $n \times n$), **sigma** - среднеквадратичное отклонение распределения Гаусса, которое используется при формировании маски **h**, параметр **a**, управляющий соотношением между центральным и граничными элементами маски, устанавливается в диапазоне $[0, 1]$, по умолчанию равен 0.2.

. Возможные варианты функции **fspecial**:

- $h = \text{fspecial}('gaussian', n, \text{sigma})$ возвращает маску **h** фильтра нижних частот Гаусса. По умолчанию n равно 3×3 , а **sigma** равно 0.5.
- $h = \text{fspecial}('sobel')$ возвращает маску фильтра Собела для выделения горизонтальных границ

$$h = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

для выделения вертикальных границ достаточно транспонировать данную маску **h**.

- $h = \text{fspecial}('prewitt')$ возвращает маску фильтра Превита для выделения горизонтальных границ

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

- $h = \text{fspecial}('laplacian', a)$ возвращает маску **h** ВЧ фильтр Лапласа. Размер маски 3×3 .
- $h = \text{fspecial}('log', n, \text{sigma})$ возвращает маску **h** фильтра, аналогичного последовательному применению фильтров Гаусса и Лапласа, так называемого лапласиана – гауссиана, **n** и **sigma** по умолчанию устанавливаются равным 5×5 и 0,5 соответственно.
- $h = \text{fspecial}('average', n)$ возвращает маску **h** усредняющего НЧ фильтра. По умолчанию размер маски **n** устанавливается равным 3×3 .
- $h = \text{fspecial}('unsharp', a)$ возвращает маску **h** фильтра, повышающего резкость изображения. Размер маски 3×3 .

Двумерная АЧХ

freqz2(h,n1,n2)

[H,f1,f2]=freqz2(h,n1,n2)

Функция **freqz2(h,n1,n2)** без выходных параметров выводит в текущее окно на экране двумерную АЧХ.

Функция **[H,f1,f2]=freqz2(h,n1,n2)**; формирует матрицу **H** размера $n1 \times n2$, которая является АЧХ на частотах, содержащихся в векторах **f1** и **f2** линейного двумерного КИХ фильтра с маской **h**.

Пример 1. Вывод АЧХ фильтра, повышающего резкость изображения.

```
>> h=fspecial('unsharp');
>> [H,f1,f2]=freqz2(h);
>> figure, mesh(f1,f2,abs(H))
```

Формирование маски линейного фильтра по желаемой АЧХ

```
h=fsamp2(f1,f2,H)
```

Функция формирует маску h линейного двумерного КИХ фильтра, основываясь на желаемой АЧХ двумерного фильтра H , для частот, передаваемых в векторах $f1$ и $f2$.

Пример. Повышение резкости изображения с помощью линейного КИХ фильтра, сформированного по желаемой АЧХ.

Получение нормализованных значений частот

```
>> [f1,f2]=freqspace(15,'meshgrid');
```

Формирование желаемой АЧХ, как функции расстояния от начала координат.

Для вычисления расстояния используется т. н. метрика городских кварталов.

```
>> dist=abs(f1)+abs(f2);
```

```
>> H=dist/max(dist(:));
```

Вывод на экран желаемой АЧХ.

```
>> mesh(f1,f2,H),colormap(cool(32));
```

Формирование маски КИХ фильтра.

```
>> h=fsamp2(f1,f2,H,[5 5])
```

Вывод на экран получившейся АЧХ.

```
>> figure,colormap(cool(32)),freqz2(h);
```

Чтение изображения и вывод его на экран.

```
>> [i,map]=imread('c:\athena.bmp');
```

```
>> figure,imshow(i);
```

Фильтрация изображения

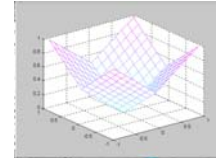
```
>> i1=mat2gray(filter2(h,im2double(i)));
```

```
>> figure,imshow(i1);
```

Контрастирование результата фильтрации

```
>> i2=imadjust(i1,[0 0.5],[]);
```

```
>> figure,imshow(i2);
```

**Формирование маски линейного фильтра методом преобразования частот**

h=ftrans2(b)

Функция формирует маску h линейного двумерного КИХ-фильтра, используя метод преобразования частот для трансформации одномерного КИХ-фильтра с коэффициентами b . Для преобразования частот используется специальная матрица трансформации t . По умолчанию применяется матрица Мак-Клеллана:

$$t = \frac{1}{8} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & -4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

В функции $h=ftrans2(b,t)$ можно указать иную матрицу трансформации.

Пример. Размытие изображения с целью подавления муара с помощью линейного двумерного КИХ, сформированного из одномерного ФНЧ методом преобразования частот:

Создание одномерного КИХ ФНЧ 14-ого порядка с частотой среза 0.2

```
>> b=fir1(14,0.2); % проектирование КИХ фильтров методом взвешивания
```

```
>> freqz(b,1,256)
```

Формирование двумерного фильтра из одномерного.

```
>> h=ftrans2(b);
```

```
>> figure,freqz2(h)
```

Чтение исходного изображения из файла и его обработка

```
>> [S,map]=imread('c:\Athena.bmp');
```

```
>> I=ind2gray(S,map);
```

```
>> I=im2double(I);
```

```
>> figure,imshow(I)
```

```
>> I1=filter2(h,I);
```

```
>> figure,imshow(I1)
```

Обобщенный нелинейный фильтр

`D=nlfilter(S,[m n],fun,P1,P2, ...)`

`D=nlfilter(X,'indexed', ...)`

Функция `D=nlfilter(S,[m n],fun,P1,P2, ...)` используется для обработки бинарных и полутоновых изображений. Формирует новое изображение, используя для фильтрации маску размером `[m n]` и функцию, осуществляющую фильтрацию, `fun`. `P1,P2` - возможные параметры, передаваемые в `fun`.

Функция `D=nlfilter(X,'indexed', ...)` аналогична предыдущей функции, но предназначена для обработки палитровых изображений.

Пример. Создание функции для фильтрации импульсного шума, передаваемой в аргументе `fun` функции обобщенного нелинейного фильтра и использование этого фильтра для удаления импульсного шума в изображении.

Фильтрация импульсного шума осуществляется операцией усреднения с порогом, которая состоит в том, что центральному пикселу в пределах маски присваивается среднее значение яркости всех пикселов в пределах маски в том случае, если разница между исходным значением центрального пиксела и средним больше заданного порогового значения. Для создания этой функции с именем 'AverageWithTh' в редакторе m-файлов надо:

1. создать новый m-файл командой

`File/New/m-file`

2. ввести команды функции:

```
function R=AverageWithTh(x,Th); % заголовочная строка
```

```
[r c]=size(x);
```

```
n=r*c;
```

```
r=floor((r+1)/2); c=floor((c+1)/2);
```

```
s=sum(x(:))/n;
```

```
if (abs(x(r,c)-s))>Th
```

```
R=s;
```

```
else
```

```
R=x(r,c);
```

```
end;
```

```
end;
```

3. Создать новый подкаталог в корневом каталоге. В качестве имени взять свои инициалы в латинском регистре.

4. Сохранить созданную функцию в файл с одноименным названием.

5. Для нахождения функции системой надо установить путь командой:

```
>> path(path,'C:\mylib');
```

Далее приведены команды по чтению исходного изображения из файла, зашумлению изображения и его фильтрация.

```
>> [X,map]=imread('c:\Athena.bmp');
```

```
>> I=ind2gray(X,map);
```

```
>> I=Im2double(I);
```

```
>> figure,imshow(I)
```

```
>> I1=imnoise(I,'salt & pepper');
```

```
>> figure,imshow(I1)
```

```
>> I2=nlfilter(I1,[3 3],'AverageWithTh',0.2);
```

```
>> figure,imshow(I2)
```

```
>> figure,imshow(c)
```

Лекция №17

Корреляционный анализ

Корреляция анализ позволяет получить на практике представление о некоторых свойствах изображения, о скорости изменения интенсивности по координатам, о протяженности однородных участков без разложения их на гармонические составляющие.

Смысл корреляционного анализа состоит в количественном измерении *степени сходства* различных сигналов. Для этого служат корреляционные функции, рассмотренные ниже.

Корреляционная функция

Корреляционная функция (КФ; английский термин — correlation function, CF) детерминированного сигнала с конечной энергией представляет собой интеграл (в бесконечных пределах) от произведения двух копий сигнала, сдвинутых друг относительно друга на время τ :

$$B_s(\tau) = \int_{-\infty}^{\infty} s(t)s(t-\tau)dt.$$

Корреляционная функция показывает степень сходства между сигналом и его сдвинутой копией — чем больше значение корреляционной функции, тем это сходство сильнее. Кроме того, корреляционная функция обладает следующими свойствами:

1 Значение КФ при $\tau = 0$ равно *энергии* сигнала, то есть интегралу от его квадрата:

$$B_s(0) = \int_{-\infty}^{\infty} s^2(t)dt = E$$

2 КФ является четной функцией своего аргумента τ :

$$B_s(\tau) = B_s(-\tau)$$

3 Значение КФ при $\tau = 0$ является максимально возможным значением:

$$B_s(\tau) \leq B_s(0)$$

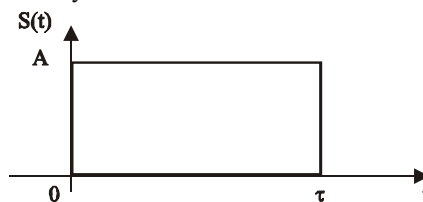
4 С ростом абсолютного значения τ КФ сигнала с конечной энергией затухает:

$$\lim_{|\tau| \rightarrow \infty} B_s(\tau) = 0$$

5 Если сигнал $s(t)$ не содержит особенностей в виде дельта-функций, его КФ не может иметь разрывов (то есть обязана быть непрерывной функцией).

В качестве примера вычислим КФ прямоугольного импульса:

$$s(t) = \begin{cases} A, & 0 \leq t \leq \tau \\ 0, & t < 0, \quad t > \tau \end{cases}$$



при $0 \leq \tau \leq T$

$$B_s(\tau) = \int_{\tau}^T A^2 dt = A^2(T - \tau)$$

при $-T \leq \tau < 0$

$$B_s(\tau) = \int_0^{T+\tau} A^2 dt = A^2(T + \tau)$$

при $|\tau| > T$

$$B_s(\tau) = 0.$$

Объединяя результаты, можно записать

$$B_s(\tau) = \begin{cases} A^2(T - |\tau|), & |\tau| \leq T \\ 0, & |\tau| > T \end{cases}$$

График КФ прямоугольного импульса показан на рис. 8

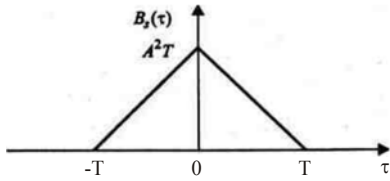


Рис. 8 Корреляционная функция прямоугольного импульса

Для изображения в качестве такой характеристики широко используется автокорреляционная функция. Для детерминированного изображения $L(x,y)$ автокорреляционная функция определяется как

$$B_L(\tau_x, \tau_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} L(x,y)L(x+\tau_x, y+\tau_y) dx dy$$

Взаимная корреляционная функция

Если КФ показывает степень сходства между сдвинутыми копиями *одного и того же* сигнала, то взаимная корреляционная функция (ВКФ; английский термин — cross-correlation function, CCF) позволяет измерить аналогичную величину для сдвинутых экземпляров двух *разных* сигналов.

Общий вид формулы КФ сохраняется, но под интегралом стоит произведение двух разных сигналов, один из которых задержан на время τ

$$B_{12}(\tau) = \int_{-\infty}^{\infty} s_1(t)s_2(t-\tau) dt$$

Очевидно, что КФ является частным случаем ВКФ, когда оба сигнала одинаковы: $s_1(t)=s_2(t)=s(t)$.

Коэффициент корреляции

При наличии статистической связи между сигналами свойства каждого из них зависят от значения, принимаемого другим сигналом. Эта связь может быть сильной или слабой, линейной или нелинейной. Мерой линейной статистической связи является коэффициент корреляции. Для дискретных изображений он вычисляется следующим образом:

$$r_{1,2} = \frac{\sum_x \sum_y (A(x,y) - A_m) * (B(x,y) - B_m)}{\sqrt{(\sum_x \sum_y (A(x,y) - A_m)^2) * (\sum_x \sum_y (B(x,y) - B_m)^2)}}$$

где $A(x,y)$, $B(x,y)$ – матрицы с изображениями,
 A_m , B_m – среднее значение матрицы.

Лекция №18

Функции корреляционного анализа в системе MatLab**Вычисление среднего значения элементов матрицы**

`m=mean2(S)`

Функция `m=mean2(S)` вычисляет среднее значение элементов матрицы `S`.

Вычисление среднеквадратического отклонения элементов матрицы

Функция `d=std2(S)` вычисляет среднеквадратическое отклонения элементов матрицы `S`. Данная функция эквивалентна `d=std(S(:))`

Вычисление коэффициента корреляции между двумя матрицами

`k=corr2(A,B)` матрицы `A` и `B` должны быть одинакового размера.

Вычисление двумерной взаимной корреляционной функции

`c=xcorr2(a,b)` – вычисляет двумерную взаимную корреляционную функцию между изображениями `a` и `b`.

`c=xcorr2(a)` - вычисляет автокорреляционную функцию и эквивалентна функции `c=xcorr2(a,a)`.

Создание видеопоследовательности

`Mov=immovie(MX, map)`

Функция формирует матрицу `Mov` из многокадрового изображения `MX` с палитрой `map`.

Многокадровое **палитровое** изображение `MX` состоит из нескольких **палитровых** изображений **одинакового** размера и использующих **одинаковую** палитру с помощью функции конкатинации (объединения) `cat`:

`MX=cat(N,F1,F2, ...)` где `F1,F2 ...` - палитровые изображения, `N`- общее количество палитровых изображений.

Для демонстрации видеопоследовательности используется функция `movie`:

`movie(Mov,N,FPS)` – показывает видеопоследовательность `Mov` `N` раз с частотой `FPS` кадров в секунду (по умолчанию `FPS=12`). Если `N` имеет отрицательное значение, то видеопоследовательность `Mov` будет показываться один раз в прямом направлении просмотра и один раз в обратном.

Пример. Создание видеопоследовательности, воспроизведение ее на экране, вычисление двумерных взаимных корреляционных функций между третьим и четвертым кадрами и построение графиков этих функций.

```
>> [D1, map]=imread('c:\earth.bmp');
```

```
>> D2=imnoise(D1,'salt & pepper');
```

```
>> h1=fspecial('unsharp');
```

```
>> D3=filter2(h1,D2);
```

```
>> h2=fspecial('prewitt');
```

```
>> D4=filter2(h2,D2);
```

```
>> MX=cat(4,D1,D2,D3,D4);
```

```
>> mov=immovie(MX,map);
```

Перед просмотром лучше предварительно закрыть все графические окна

```
>> movie(mov,5,2)
```

```
>> c=xcorr2(D1,D4);
```

Распознавание объектов на изображении

Методы автоматизированного распознавания объектов по их изображениям является в некотором смысле универсальной методологией обработки и оценивания информации и выяснения скрытых закономерностей. Распознавание трехмерных объектов по их двумерным изображениям стало одной из важнейших задач анализа сцен и машинного зрения. Исходную для распознавания информацию содержат изображения в различных частях полного спектра излучений (оптические, инфракрасные, ультразвуковые и т.д.), полученные различными способами (телевизионные, фотографические, лазерные, радиолокационные и т.д.), преобразованные в цифровую форму и представленные в виде числовой матрицы. Под объектом понимается не только (и не столько) цифровое представление локального сегмента двумерной сцены, а некоторое его приближенное

описание, в виде набора характерных свойств (признаков). Основное назначение описаний (образов объектов) – это их использование в процессе соответствия, осуществляемого путем сравнения (соответствия). Задачей распознавания является определение "скрытой" принадлежности" объектов к тому или иному классу путем анализа вектора значений наблюдаемых признаков. Информацию о связи между значениями признаков объекта и его принадлежностью к определенному классу алгоритм распознавания должен извлечь из обучающей совокупности объектов, для которых известны либо значения и признаков и классов, либо только значения признаков. В первом случае задача называется задачей обучения распознавания образов с учителем, а во втором – без учителя.

При отнесении (классификации) объектов требуется применить некоторое установленное ранее правило, чтобы решить, какому образу (классу) принадлежит объект. В задаче *расознавания с обучением* правило классификации должно вырабатываться на основе исследования множества объектов с известной принадлежностью различным классам (обучающего множества или выборки). В задаче *автоматического формирования образов* объекты предъявляются "наблюдателю" без указания их принадлежности классам (расознавание образов без учителя).

Наблюдатель (алгоритм распознавания) в этом случае должен сам построить соответствующее определение классов (кластерный анализ). При этом предполагается уже решенной задача сегментации, т. е. определение границ фрагментов, каждый из которых допустимо рассматривать как единое целое (объект).

Исследования по распознаванию образов пространственных объектов отличаются большим разнообразием в постановке задач и выборе средств их решения (методов обработки соответствующих фрагментов изображений), что является следствием разнообразия областей практического применения. Традиционными задачами распознавания являются задачи обнаружения и распознавания объектов, имеющих заданную форму на основе зашумленных и (возможно) деформированных изображений, например, задача идентификации и распознавания человеческих лиц (системы охраны, верификация кредитных карточек, криминалистическая экспертиза, на телеконференциях). Для ее решения предложены методы распознавания, основанные на теории нейрокомпьютерных сетей, корреляционно-экстремальных алгоритмах, методах вычисления статистических и алгебраических моментов, контурном анализе, 3D-моделировании и др.)

В задачах мониторинга (контроля) природной среды по аэрокосмическим изображениям наметилось еще несколько важных подходов к построению информативных признаков. В частности, когда были получены первые многозональные и спектральнозональные данные дистанционного зондирования, большинство разработчиков машинных методов интерпретации обратились к изучению спектральных свойств природных объектов, поскольку яркости соответствующих им элементов изображений в различных спектральных диапазонах позволяли идентифицировать их при относительно низких вычислительных затратах. Кроме того, выяснилось, что при распознавании природных образований оказались чрезвычайно важными также признаки, характеризующие локальную пространственную организацию полутонов (текстуру) объектов анализа.

Выделение признаков изображения

Признаком изображения называется его простейшая отличительная характеристика или свойство. Некоторые признаки являются естественными в том смысле, что они устанавливаются визуальным анализом изображения, тогда как другие, так называемые искусственные признаки, получаются в результате его специальной обработки или измерений. К естественным признакам относятся яркость и текстура различных областей изображения, форма контуров объектов. Гистограммы распределения яркости и спектры пространственных частот – это примеры искусственных признаков.

Яркостные признаки

Наиболее важным признаком изображения является светлота. Светлота выражается через такие величины, как яркость, спектральная интенсивность излучения, координаты цвета, которые называются яркостными признаками. Измерение яркостных признаков можно проводить либо в отдельных точках изображения, либо в их окрестностях. Например, средняя яркость окрестности точки (j, k) изображения размером $(2W + 1) \times (2W + 1)$ элементов определяется как

$$\bar{Y}(j, k) = [1/(2W + 1)^2] \sum_{m=-W}^W \sum_{n=-W}^W Y(j + m, k + n)$$

Существует множество различных способов определения яркостных признаков. Можно использовать значения яркости или координат цвета непосредственно либо перейти к новым яркостным признакам, выполнив некоторое линейное, нелинейное или необратимое преобразование. Измерение яркостных признаков приобретает особую важность при выделении изображенных объектов и при маркировке таких объектов.

Гистограммные признаки

Гистограммы можно использовать для создания некоторого класса признаков изображения. Форма гистограммы распределения частот дает много сведений о свойствах изображения. Например, узкая гистограмма указывает на низкий контраст, а бимодальная гистограмма предполагает наличие областей разной яркости.

Пространственно – спектральные признаки

Спектральные коэффициенты показывают степень корреляции соответствующих базисных функций с изображением. Если базисное изображение имеет ту же пространственную форму, что и признак, который необходимо обнаружить на изображении, то обнаружение признака можно выполнить просто путем наблюдения значения соответствующего спектрального коэффициента. Практическая сложность состоит в том, что объекты, которые необходимо обнаружить, часто имеют сложную форму и распределение яркостей и, следовательно, не соответствуют точно более простым яркостным образам, которые представляются базисными функциями большинства преобразований.

Выделение признаков, представленных в виде спектральных коэффициентов, было исследовано в различных практических задачах, в которых эти признаки использовались в качестве входных данных для системы распознавания образов. Спектральные признаки находят широкое применение – от классификации земельных ресурсов до диагностики болезней по рентгеновским снимкам.

Контурные признаки

Резкие изменения яркости, координат цвета или параметров, характеризующих текстуру, являются важными простейшими признаками, поскольку они часто определяют очертания изображенных объектов. Локальные разрывы значений яркости называются яркостными перепадами, или яркостными контурами.

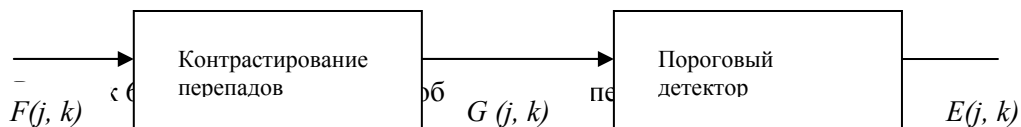
Общий подход к обнаружению перепадов на одноцветном изображении показан в виде блок-схемы на рисунке 6. Исходное изображение, представленное массивом чисел $F(j, k)$, подвергается линейной или нелинейной обработке с тем, чтобы усилить перепады яркости. В результате образуется массив чисел $G(j, k)$, описывающий изображение с подчеркнутыми измерениями яркостей. Затем выполняется операция сравнения с порогом и определяется положение элементов изображения с ярко выраженными перепадами. Если

$$G(j, k) < T_L(j, k),$$

то имеет место нисходящий перепад, а при

$$G(j, k) < T_U(j, k)$$

- восходящий перепад. Величины $T_L(j, k)$ и $T_U(j, k)$ представляют собой нижнее и верхнее пороговые значения.



Выбор порога является одним из ключевых вопросов выделения перепадов. При слишком высоком уровне порога не будут обнаружены структурные элементы с низким контрастом. Наоборот, слишком низкий уровень порога явится причиной того, что шум будет ложно принят за перепад. Для обозначения положения перепадов на изображении часто формируют контурный препарат – массив элементов $E(j, k)$. Например, положение точек восходящих препаратов можно было бы отмечать белыми точками на черном фоне. Можно также точки восходящих препаратов отмечать белым цветом, нисходящих – черным, а остальные элементы изображения – некоторым средним уровнем яркости.

Цветовые контуры

Цветные изображения можно описывать количественно, определяя для каждого элемента координаты цвета T_1 , T_2 , T_3 , пропорциональные количеству красного, зеленого и синего основных цветов, необходимых для уравнивания цвета этого элемента. Яркость элемента представляет собой взвешенную сумму значений координат цвета

$$Y = \alpha_1 T_1 + \alpha_2 T_2 + \alpha_3 T_3,$$

где α , — весовые коэффициенты.

Существует несколько определений цветовых контуров. Можно считать, что контур на цветном изображении соответствует только перепаду яркости. Однако это определение исключает контуры, связанные с резкими изменениями цветового тона и насыщенности в областях с постоянной яркостью. Другой подход состоит в том, что исследуется каждая из трех компонент изображений, и считается, что цветовой контур существует, если яркостный перепад обнаружен одновременно на всех компонентах. Третье определение состоит в том, что контур имеет место, если расстояние между векторами цвета по обе стороны предполагаемого перепада превосходит некоторый пороговый уровень. В двух последних случаях определений цветовых контуров результаты обнаружения полностью зависят от выбранной системы координат цвета.

Тесты и задания по курсу ЦОИ

Тест по способам организации информации в системе MatLab и визуализации 2-D и 3-D графиков.**Задание 1**

1. Создать вектор-строку, если начальный элемент равен 0, конечный 14, шаг равен 1. Транспонировать строку в столбец.
2. Создать три вектора-строки из 5-ти элементов многочлена $f = [x^n, x^{n-1}, x^{n-2}, x^{n-3}, x^{n-4}]$ для $n=5$ и $x=2$; $x=3$; $x=4$. Объединить эти строки в матрицу $A(3 \times 5)$.
3. Создать три вектора-столбца из 5-ти элементов арифметической прогрессии $a_n = a_1 + d(n-1)$ для $a_1=2$ $d=1$; $a_1=7$ $d=2$; $a_1=10$ $d=-2$. Объединить эти столбцы в матрицу $B(5 \times 3)$.
4. Создать матрицы размерностью 3×3 : C - единиц; D - нулей; E - равномерно распределенных случайных чисел; F - нормально- распределенных случайных чисел
5. Транспонировать матрицу B и объединить с матрицей A в матрицу $M(6 \times 5)$
6. Из матрицы A убрать вторую строку.
7. У матрицы B обнулить третью строку и убрать две последние строки.
8. Создать с помощью функции `hermat` матрицу, состоящую из 2×3 матриц B .
9. Создать символьные константы: 1 - миру мир; 2 – введите матрицу, Ввести комментарий: использование интерактивного ввода.
10. Создать структуру, содержащую матрицы размерностью 3×3 и ввести с клавиатуры три матрицы

1 1 1	1 1 1	-1 1 1
1 -2 1	-1 -2 1	-1 -2 1
-1 -1 -1	-1 -1 1	-1 1 1

Записать созданные матрицы в файлы на диск.

11. Определить максимальное и минимальное значения матрицы F из пункта 4 двумя способами: с помощью специальной функции `reshape` и через цикл `While`.

Задание 2

1. Построить на отрезке $[-1, -0.3]$ с шагом 0.005 графики функций

$$f1 = \sin \frac{1}{x^{-2}}, f2 = \sin \frac{1.2}{x^{-2}}$$

Первый график вывести красной сплошной линией, а второй – зеленой штрих-пунктирной линией с маркерными точками x .

В этом же окне построить эти графики с помощью функции вывода графика дискретных отсчетов без затирания предыдущего результата.

2. Построить графики суточных температур :

значения времени	time = 0 4 7 9 10 11 12 13 13.5 14 14.5 15 16 17 18 20 22
значения температуры 10 мая:	tem1= 14 15 14 16 18 17 20 22 24 28 25 20 16 13 13 14 13
значения температуры 11 мая:	tem2= 12 13 13 14 16 18 20 20 25 25 25 20 16 12 12 11 10

 Оформить графики заголовком 'All temperature', по оси x подписать 'Time (hour)', по оси y 'Temperature (C)', в легенде – '10 may', '11 may' и разместить ее в нижнем левом углу.
3. Построить 3-х мерные графики функции

$$z(x, y) = 4 \cdot \sin 2\pi x \cdot \cos 1.5\pi y \cdot (1 - x^2) \cdot y \cdot (1 - y)$$
 на прямоугольной области $x \in [-1, 1], y \in [0, 1]$ с шагом 0.05 всеми способами, рассмотренными в лабораторной работе, размещая их в отдельных областях окна. Названия команд, применяемых для построения графиков, включить в заголовки подграфиков.
4. Построить в полярной системе координат графики функций:

для $\varphi \in [-\pi, \pi]$ с шагом 0.1

 - $r = 4 \cdot \cos(\varphi) - 2$ улитка Паскаля

- $r = \sqrt{8 \cdot \cos(2 \cdot \varphi)}$ лемниската
- в одном окне построить графики четырех лепестковой розы, меняя значение параметра a от 8 до 14 с шагом 2:
 $r = a \cdot |\sin(2 \cdot \varphi)|$

Тест по получению характеристик линейных систем

Задание 1

1. Рассчитать выходной сигнал на выходе устройства с импульсной характеристикой, равной 3 6 9 при входном сигнале 5 6 7. Вывести его на экран. Очистить вектор с импульсной характеристикой и рассчитать ее, на основе заданного входного и вычисленного выходного сигналов, используя соответствующую функцию Matlab. Вывести дискретные отсчеты импульсной характеристики на экран.

2. Получить графики АЧХ и ФЧХ устройства со следующей функцией передачи:

$$H = \frac{0.2S^2 + 0.3S + 1}{S^2 + 0.4S + 1}$$

Построить линейную АЧХ и ФЧХ без лишних разрывов.

3. Рассчитать фильтры-прототипы Чебышева, эллиптический, задав следующие значения параметров: порядок фильтра $n=5$, величина пульсаций в полосе пропускания $R_p=0.5$, величина пульсаций в полосе задерживания $R_s=20$. Получить графики АЧХ и ФЧХ.

Тест по геометрическим и амплитудным преобразованиям изображений

Задание 1

Выполнить следующие действия:

1. Вырезать из палитрового изображения, взятого из файла Chip.bmp, фрагмент шириной 95 и высотой 60, начиная с координат $x=111$, $y=34$, показать исходное изображение и фрагмент. Увеличить полученный фрагмент в 1.5 раза и повернуть против часовой стрелки на 60° . Показать результат.
2. Увеличенный фрагмент сделать ярче, и вывести результат и исходное изображение в одно окно для сравнения.
3. Преобразовать исходное изображение в полутоновое. Построить гистограмму. Выполнить его линейное контрастирование, определив исходный диапазон по гистограмме. Вывести контрастированное и исходное полутоновое изображение в одно окно.
4. Получить негатив исходного полутонового изображения.
5. Выполнить контрастирование исходного полутонового изображения с гамма-коррекцией – получить более светлое и более темное изображения. Вывести результат и исходное изображение в одно окно для сравнения.
6. Построить гистограммы для этих трех изображений и вывести результат в одно окно.
7. Выполнить эквализацию палитрового изображения из файла Technlgy.bmp. Построить гистограммы исходного палитрового изображения и эквализированного изображения. Результат вывести в одно окно (в верхнем ряду – изображения, под ними их гистограммы).
8. Прочитать палитровое изображение Technlgy.bmp, Получить два изображения, уменьшая количество цветов до $n=7$ и с $tol=0.7$. Результат вывести на экран в разные окна. Преобразовать полученные изображения в полутоновые и создать по ним изображения из четырех уровней яркости. Вывести на экран.

Тест по фильтрации изображений

Задание 1

Выполнить следующие действия:

1. Прочитать палитровое изображение из файла `Earth.bmp`, вывести на экран, преобразовать в полутоновое, добавить импульсный шум, отфильтровать зашумленное изображение медианной фильтрацией и обобщенным нелинейным фильтром. Зашумленное и отфильтрованные изображения вывести в одном окне для сравнения. Задание оформить в виде функции. В качестве входного аргумента передавать название файла с палитровым изображением.

2. Выполнить фильтрацию:

- а) Отфильтровать изображение с помощью масок кругового градиента. Ввод масок двумерных линейных фильтров, построение их АЧХ, фильтрацию изображения с их помощью и вывод результата выполнить в цикле. Маски двумерных линейных фильтров:

север	северо-восток	восток	юго- восток
1 1 1	1 1 1	-1 1 1	-1 -1 1
1 -2 1	-1 -2 1	-1 -2 1	-1 -2 1
-1 -1 -1	-1 -1 1	-1 1 1	1 1 1
юг	юго-запад	запад	северо- запад
-1 -1 -1	1 -1 -1	1 1 -1	1 1 1
1 -2 1	1 -2 -1	1 -2 -1	1 -2 -1
1 1 1	1 1 1	1 1 -1	1 -1 -1

- в) Отфильтровать изображение с помощью масок лапласиановских фильтров

0 -1 0	-1 -1 -1	-1 -2 -1
-1 -4 -1	-1 8 -1	-2 4 -2
0 -1 0	-1 -1 -1	-1 -2 -1

- Прочитать палитровое изображение из файла Technlgy.bmp, вывести на экран, преобразовать в полутоновое. Получить маску оператора Превитт. Выполнить фильтрацию исходного полутонового фрагмента маской оператора Превитт отдельно по горизонтали и по вертикали и вместе на одном изображении;
- Прочитать палитровое изображение из файла 'Construc.bmp', вывести на экран, преобразовать в полутоновое. Взять в качестве АЧХ фильтра функцию расстояния от начала координат, сформировать по АЧХ маску фильтра и отфильтровать полутоновое изображение.
- Выполнить эквализацию полутонового изображения, полученного из палитрового изображения, хранящегося в файле earth.bmp. Найти двумерную взаимную корреляционную функцию между исходным и полученными изображениями. Получить график.
- Вывести АЧХ всех фильтров, создаваемых функцией по заданию масок предопределенного фильтра, в одном окне с заголовками для каждого фильтра.