КАЗАНСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ФИЗИКИ

Кафедра радиофизики

П. А. Корчагин

ЦИФРОВАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ

Часть 1

Геометрические преобразования

Методические указания к выполнению лабораторных работ

Казань – 2016

УДК 621.391(076)

К70

Принято на заседании кафедры радиофизики Протокол № 8 от 25 марта 2015 года

Рецензент

кандидат физико-математических наук,

доцент кафедры радиоастрономии КФУ Е.Ю. Зыков

К70 Корчагин П. А.

Цифровая обработка изображений. Часть 1. Геометрические преобразования. Методические указания к выполнению лабораторных работ. – Казань: Казан. ун-т, 2016. – 17 с.

Лабораторные работы посвящены практическому освоению алгоритмов геометрических преобразований на плоскости и изучению базовых функций Image Processing Toolbox (IPT) пакета MATLAB.

Пособие рассчитано на студентов старших курсов, обучающихся по направлениям «Информационная безопасность» и «Радиофизика и электроника» и содержит, необходимые для выполнения лабораторных работ, теоретические и справочные сведения.

Содержание

Введение 4

Лабораторная работа № 1 5

Базовые операции ввода-вывода цифровых изображений 5

Лабораторная работа № 2 8

Функции геометрического преобразования изображений 8

Лабораторная работа № 3 11

Геометрические преобразования векторных изображений 11

ПРИЛОЖЕНИЕ 1 16

Список литературы 17

Введение

Области использования цифровой обработки в настоящее время значительно расширяются, вытесняя аналоговые методы обработки сигналов изображений. Методы цифровой обработки широко применяются в промышленности, искусстве, медицине, науке. Они решают задачи управления процессами, автоматизации обнаружения и сопровождения объектов, распознавания образов, визуализации, биометрической идентификации.

Изучение основ цифровой обработки изображений, представляет интерес для студентов направлений подготовки: «Информационная безопасность», «Радиофизика и электроника», как будущим разработчикам и пользователям информационных систем.

В качестве программного инструмента для выполнения работ был выбран IPT пакета MATLAB, который занимает ведущие позиции в образовательной и индустриальной сфере.

Лабораторные работы направленны на изучение основных графических функций IPT и алгоритмов геометрических преобразований изображений на плоскости.

В методических указаниях приводятся краткие теоретические и справочные сведения по каждой работе и описан ход выполнения работы.

Результаты выполнения лабораторных работ оформляются в виде отчета, который должен содержать скрины этапов работы, листинги M – функций, результаты тестирования и выводы. Титульный лист отчета должен содержать фразу: “Отчет по лабораторной работе (название работы) студента группы (номер группы) (Фамилия, инициалы)”. Внизу листа следует указать текущий год. Отчет сдается преподавателю в электроном виде в формате PDF.

Лабораторная работа № 1

Базовые операции ввода-вывода цифровых изображений

Цель работы: Освоение операций загрузки цифровых изображений, работы с графическими форматами файлов и вывода изображений на экран.

1. Скачайте цифровое изображение с бесплатного фотобанка. Можно воспользоваться одним из сервисов, на которые распространяется действие публичных лицензий:

Pixabay - https://pixabay.com;

Stockvault - http://www.stockvault.net;

Gratisography - http://www.gratisography.com.

Переименуйте файл задав имя im1.jpg и переместите его в папку MATLAB локального пользователя.

2. Выполните загрузку изображения в среду MATLAB. Для этого наберите команду:

Im= imread('C:\Users\user1\Documents\MATLAB\im1.jpg'); .

Справочные сведения

Функция imread(filename, fmt) читает из файла с именем filename бинарное, полутоновое или полноцветное изображение и помещает его в массив. Параметр fmt в вызове функции может быть опущен, в этом случае формат файла автоматически определяется из его содержимого. Допустимые форматы файла: BMP, TIFF, JPEG, PCX, HDF, XWD.

3. Выполните вывод изображения на экран с помощью команды:

imshow(Im); .

Справочные сведения

Функция imshow(Im) выводит на экран полноцветное изображение Im.

4. Сохраните изображение в файл формата JPEG с максимальным, средним и минимальным степенями сжатия. Используйте команды:

imwrite(Im,'im0.jpg ','Quality',0);

imwrite(Im,'im50.jpg ','Quality',50);

imwrite(Im,'im100.jpg ','Quality',100); .

Справочные сведения

Функция imwrite(Im, filename, fmt) записывает в файл с именем filename бинарное, полутоновое или полноцветное изображение Im. Функция imwrite(Im, map, filename, fmt) записывает в файл с именем filename палитровое изображение Im с палитрой map. Формат файла определяется параметром fm. Если запись осуществляется в JPEG-файлы, то можно указывать показатель качества сжатого изображения. Для этого нужно после fmt указать 'Quality' и через запятую число, которое определяет степень сжатия изображения. Этот показатель может принимать значения в диапазоне от 0 до 100. Чем меньше значение этого показателя, тем выше степень сжатия, но хуже качество изображения.

5. Визуально проанализируйте наличие артефактов в полученных изображениях. Для этого выведите исходное и сжатые изображения в одно графическое окно.

Справочные сведения

*Функция subimage(Im) в сочетании с функцией MATLAB subplot позволяет вывести в одно окно несколько полноцветных, полутоновых и бинарных изображений Im. Функция subplot(m, n, p) разбивает текущее окно на m*m подокон и устанавливает текущим окно с номером p. Подокна нумеруются слева направо и сверху вниз, начиная от левого верхнего подокна, которое имеет номер 1. Функция subimage выводит изображение в текущее подокно.*

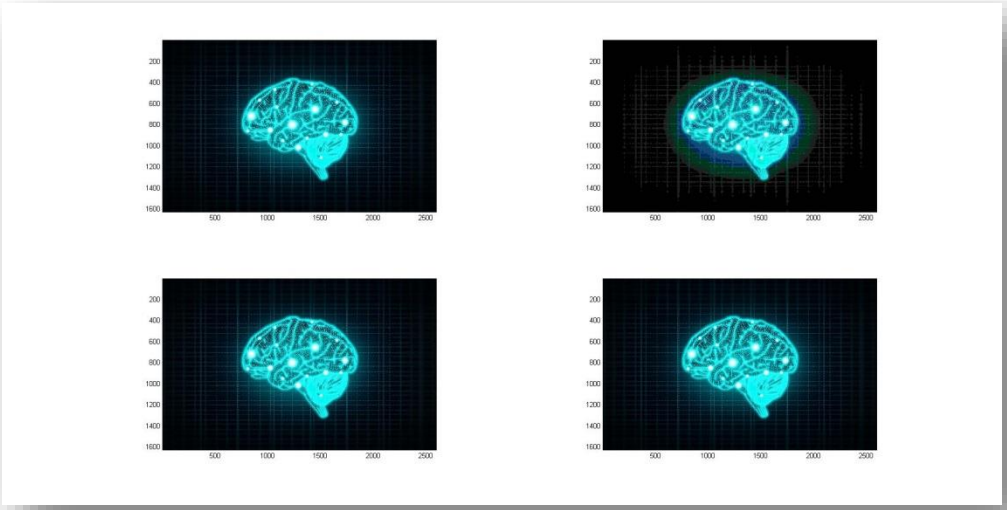

Рис. 1 Вывод изображений в одно окно

6. Разработайте M – функцию автоматизирующую выполнение заданий лабораторной работы.

Лабораторная работа № 2

Функции геометрического преобразования изображений

Цель работы: Изучение и использование стандартных функций среды MATLAB выполняющих геометрические преобразования изображений.

1. Загрузите изображение в среду MATLAB. Определите размеры изображения (число строк и столбцов).

Справочные сведения

Функция size(Im) возвращает количество строк и столбцов в изображении.

2. Выполните кадрирования изображения. Необходимо вырезать ¼ изображения, как показано на рис. 2.

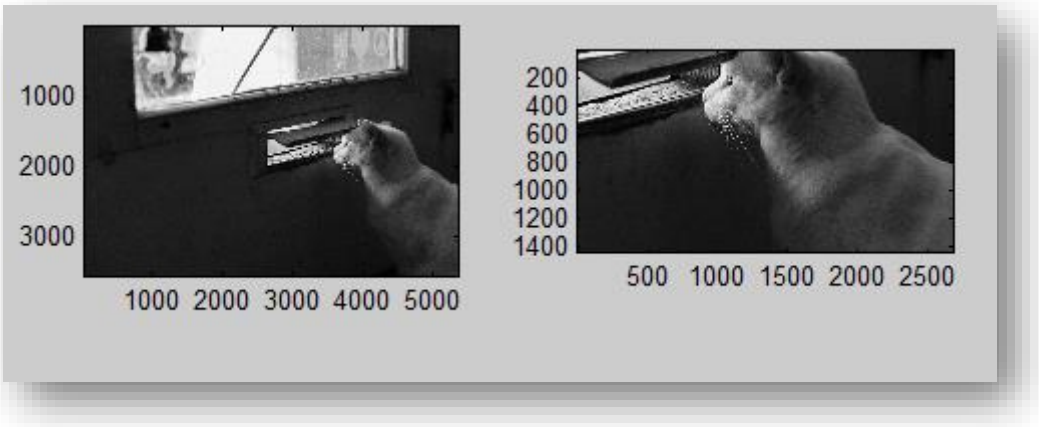

Рис. 2 Результат кадрирования изображения

Справочные сведения

Функция imcrop возвращает изображение, ограниченное заданным прямоугольником. Используя функции rIm=imcrop(Im, rect), можно явно определить ограничивающий прямоугольник, где rect - вектор из четырех элементов: [xmin ymin w h], которые задают положение левого верхнего угла (xmin ymin), ширину (w) и высоту (h) .

3. Над полученным изображением (из задания 2) выполните масштабирование с коэффициентами 4 и 0.25 и заданным методом интерполяции. Результат и исходное изображение выведите в одно графическое окно. Проанализируйте наличие артефактов.

Справочные сведения

Функция zIm=imresize(кIm, m, method) создает изображение zIm, размеры которого в m раз отличаются от размеров исходного изображения. Если m принадлежит диапазону от 0 до 1, то происходит уменьшение изображения, если m больше 1, то увеличение. 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 (установлен по умолчанию, и данный параметр может быть опущен при вызове функции);

'bilinear' - использовать интерполяцию по билинейной поверхности;

'bicubic' - использовать интерполяцию по бикубической поверхности.

4. Выполните поворот изображения на 360^0 с шагом 15^0 и заданным методом интерполяции. При выполнении задания использовать возможности анимации для демонстрации результата.

Справочные сведения

Функция rIm=imrotate(Im, angle, method) создает изображение rIm, соответствующее повернутому исходному изображению, используя один из

методов интерполяции. Метод интерполяции определяется 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.

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

Функция getframe создает фрейм для анимации.

Функция movie воспроизводит анимацию.

Пример. Необходимо анимировать масштабирование с коэффициентами:0.5,1,1.5.Предусмотреть 10 повторов анимации.

```
z=0.5;
for t=1:3
zIm=imresize(Im, z);
imshow(zIm);
z=z+0.5;
M(t)=getframe;

end

movie(M, 10)% 10  количество повторов
```

5. Разработайте М – функцию автоматизирующую выполнение заданий лабораторной работы.

Лабораторная работа № 3

Геометрические преобразования векторных изображений

Цель Изучение и программная реализация в среде MATLAB алгоритмов геометрических преобразований на плоскости (2-D преобразований).

1. Разработать и протестировать М - функцию построения случайного многоугольника для различных максимальных углов приращения Ω: 90^0, 120^0, 150^0, 180^0. Алгоритм построение приведен в Приложении 1. Координаты вершин должны храниться в матрице следующего вида:

$$\begin{pmatrix} x_1 & y_1 & 1 \\ \dots & \dots & 1 \\ x_n & y_n & 1 \\ x_1 & y_1 & 1 \end{pmatrix},$$

где n – количество вершин.

2. Построить диаграмму процентного соотношения количества вершин многоугольника от угла Ω.

Справочные сведения

Функция rand('seed', x0) начальному значению генератора случайных чисел значение x0.

Функция X = rand(n) формирует массив размера n на n*, элементами которого являются случайные величины, распределенные по равномерному закону в интервале (0, 1).*

Функция rand *без аргументов формирует случайное число, подчиняющееся равномерному закону распределения в интервале (0, 1).*

2. Разработать и протестировать М - функцию, выполняющую 2-D преобразование над случайным многоугольником. Вариант преобразования задается преподавателем (см. Таблица 1).

Таблица 1

Вариант	Преобразование	Матрица преобразования
1	Поворот относительно произвольной точки О	$\begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 0 \\ -\sin(\varphi) & \cos(\varphi) & 0 \\ X & Y & 1 \end{bmatrix}$ $X=o_x(1-\cos(\varphi))+o_y\sin(\varphi)$ $Y= o_y(1-\cos(\varphi))-o_x\sin(\varphi)$
2	Перемещение	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ dx & dy & 1 \end{bmatrix}$
3	Масштабирование	$\begin{bmatrix} zx & 0 & 0 \\ 0 & zy & 0 \\ 0 & 0 & 1 \end{bmatrix}$
4	Отражение от прямой $x=a$	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 2a & 0 & 1 \end{bmatrix}$
5	Сдвиг	$\begin{bmatrix} 1 & S2 & 0 \\ S1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

3. Произвести анимацию преобразования путем дискретизации. Результат анимации сохранить в файл формата GIF. Разработать соответствующую М – функцию.

4. Выведите информацию о полученном изображении. Для этого воспользуйтесь функцией **imfinfo**.

Справочные сведения

Функция comet(y) рисует движение точки по траектории, заданной одномерным массивом y, в виде головы и хвоста кометы.

Функция comet(x, y) рисует движение точки по траектории, заданной массивами x и y.

Функция plot(x, y) соответствует построению обычной функции, когда одномерный массив x соответствует значениям аргумента, а одномерный массив y - значениям функции. Когда один из массивов X или Y либо оба двумерные, реализуются следующие построения:

если массив Y двумерный, а массив x одномерный, то строятся графики для столбцов массива Y в зависимости от элементов вектора x;

если двумерным является массив X, а массив y одномерный, то строятся графики столбцов массива X в зависимости от элементов вектора y;

если оба массива X и Y двумерные, то строятся зависимости столбцов массива Y от столбцов массива X.

Функция rgb2ind создает палитровое изображение из полноцветного, используя один из четырех способов: запись в виде палитрового изображения без уменьшения количества цветов, установка равномерной палитры, оптимальный подбор палитры, использование некоторой предопределенной палитры.

Команда [im, map] =rgb2ind(RGB) создает палитровое изображение im из полноцветного f, составляя палитру map из всех уникальных цветов, представленных в исходном изображении. Результирующая палитра map может быть очень большого размера.

Команда [im,map] = rgb2ind(f,256) создает палитровое изображение im из полноцветного f, 256 — ограничение на количество цветов в палитре.

Функция imwrite(im,map,'test.gif','DelayTime',0,'LoopCount',inf) записывает анимацию в GIF – файл.

DelayTime определяет время задержки между кадрами анимации, LoopCount задает число повторений. LoopCount=inf зацикливает анимацию.

Функция imfinfo(filemane) возвращает информацию об изображении и способе его хранения из файла с именем filemane. Информация заносится в структуру info. Структуры info различаются для разных форматов файлов, однако первые 9 полей всегда содержат следующую общую информацию, которую можно представить в виде Таблицы 2.

Таблица 2

Имя поля	Тип	Описание
FileName	Строка	Имя файла, если файл находится в текущей директории, или полный путь к файлу
FileModeDate	Строка	Дата и время последней модификации файла
FileSize	Число	Размер файла в байтах
Format	Строка	Формат файла, совпадающий с параметром ***fmt***
FormatVersion	Строка или число	Версия формата
Width	Число	Ширина изображения в пикселях
Height	Число	Высота изображения в пикселях
BitDepth	Число	Глубина изображения в битах на пиксель
ColorType	Строка	Тип изображения: ▪ ***'truecolor'*** или ***'RGB'*** для полноцветных изображений; ▪ ***'grayscale'***для полутоновых; ▪ ***'indexed'***для палитровых

Пример. Необходимо произвести операцию сдвига над треугольном с координатами вершин: 1 1; 2 1;1 2. В матрице преобразования S1=0 и S2=5. Результат сохранить в GIF – файл с анимацией пошагового преобразования с табуляцией S2 с шагом 0.5.

```
s= 0.5:0.5:5;
Tr=[1 1;2 1;1 2;1 1];
plot(Tr(:,1),Tr(:,2));
axis( [ 0, 3, 0, 12] );
```

```
f = getframe;
[im,map] = rgb2ind(f.cdata,256);
im(1,1,1,10) = 0;
for k = 1:10
Sh=[1 s(k);0 1];
  Trs=Tr*Sh
  plot(Trs(:,1),Trs(:,2));
 axis( [ 0, 3, 0, 12] );
  f = getframe;
  im(:,:,1,k) = rgb2ind(f.cdata,map);
end
imwrite(im,map,'test.gif','DelayTime',0,'LoopCount',inf)
```

ПРИЛОЖЕНИЕ 1

Алгоритм построения многоугольника со случайным количеством вершин (не менее 3-х)

Рассмотрим алгоритм генерирования случайного многоугольника с вершинами (не менее трех) последовательно соединенными друг с другом не пересекающейся замкнутой ломанной линией со случайным направлением обхода:

1. Из заданной точки O как из центра проводим лучи по $0^0 \leq \varphi \leq 360^0$ углами к оси OX. Начальное значения угла $\varphi_1=0^0$, а последующие углы рассчитываем путем приращения $\varphi_{n+1}=\varphi_n+\Delta\varphi_n$ на случайное значение $\Delta\varphi_n = B\ rand()$. Максимальное приращения углов $120^0 \leq B \leq 180^0$выбираем с таким расчетом, чтобы минимальное число лучей было равно трем.

2. Вдоль лучей откладываем расстояния $r_n=a+(b-a)*rand()$, генерируемые как случайные числа в диапазоне от a до b. В итоге, получаем вершины многоугольника:

$$p_n = O + r_n[\cos(\varphi_n)\sin(\varphi_n)], n = 1, 2, \ldots..$$

3. Генерируем $d=2\ rand()$. При $d \geq 1$нумеруем вершины P в обратном порядке.

4. Возврат P.

Список литературы

1. Никулин Е. А. Компьютерная геометрия и алгоритмы машинной графики. – СПб.: БХВ-Петербург, 2003. – 560 с.: ил.

2. Потемкин В. Г. Справочник по MATLAB. Графические команды и

функции //Интернет–ресурс: http:// matlab.exponenta.ru/ imageprocess/ index.php (Дата обращения: 24.08.2016).

3. Р. Гонсалес, Р. Вудс, С. Эддинс. Цифровая обработка изображений в среде MATLAB. – Москва:Техносфера, 2006. – 616 с.

4. Фисенко В.Т., Фисенко Т.Ю. Компьютерная обработка и распознавание изображений: учеб. пособие. - СПб: СПбГУ ИТМО, 2008. – 192 с.

5. Р. Гонсалес, Р. Вудс. Цифровая обработка изображений. – Москва: Техносфера, 2012. – 1104 с.