

ЛР2: Основы интерполяции изображений.

Цель: Научиться изменять размеры и выполнять поворот цифровых изображений в MATLAB.

Теоретическая часть

Преобразование координат может быть выражено в форме уравнения

$(x, y) = T(v, w)$	(1.1)
--------------------	-------

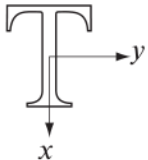
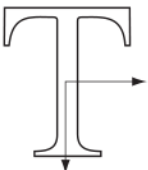

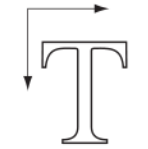
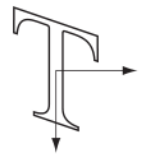
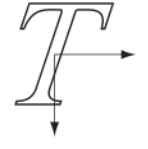
где (v, w) — координаты пикселя на исходном изображении, а (x, y) — координаты соответственного пикселя на преобразованном изображении.

Одним из наиболее часто применяемых преобразований координат является аффинное преобразование, которое имеет общую форму

$\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} v & w & 1 \end{bmatrix} \cdot T = \begin{bmatrix} v & w & 1 \end{bmatrix} \cdot \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$	(1.2)
---	-------

В зависимости от выбранных значений элементов матрицы T , это преобразование позволяет осуществлять изменение масштаба, поворот, сдвиг или скос множества координатных точек.

В таблицу указаны значения элементов матрицы, используемые для реализации этих преобразований. Реальная сила матричного представления состоит в том, что оно предлагает единый каркас для выполнения последовательности различных операций. Например, чтобы изменить размеры изображения, повернуть его и передвинуть результат в другое место, достаточно просто сформировать матрицу 3×3 , равную произведению матриц масштабирования, поворота и параллельного переноса из таблицы.

Название преобразования	Аффинная матрица T	Преобразование координат	Пример
Тождественное преобразование	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x &= v \\ y &= w \end{aligned}$	
Изменение масштаба	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x &= c_x v \\ y &= c_y w \end{aligned}$	
Поворот	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x &= v\cos\theta - w\sin\theta \\ y &= v\sin\theta + w\cos\theta \end{aligned}$	
Параллельный перенос (сдвиг)	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$\begin{aligned} x &= v + t_x \\ y &= w + t_y \end{aligned}$	
Вертикальный скос	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x &= v + s_v w \\ y &= w \end{aligned}$	
Горизонтальный скос	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{aligned} x &= v \\ y &= s_h v + w \end{aligned}$	

Интерполяция является основным инструментом, широко используемым при решении таких задач, как увеличение и уменьшение изображений, их поворот и коррекция геометрических искажений. В своей основе интерполяция — это процесс, при котором имеющиеся данные используются для оценки значений в неизвестных точках.

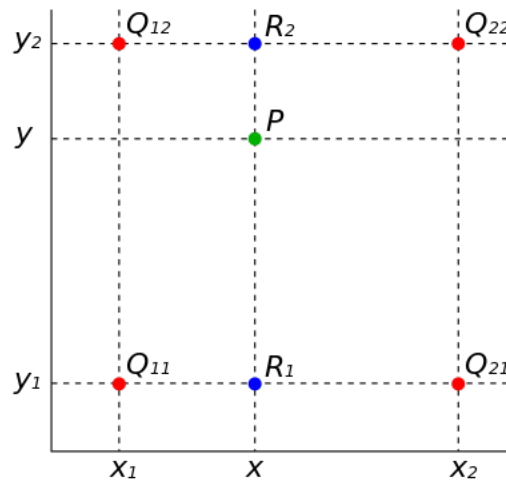
Метод билинейной интерполяции используется для увеличения изображения.

Метод присваивает яркость пикселю нового изображения на основе четыре ближайших его соседей. Пусть (x, y) — координаты точки нового изображения, которой нужно присвоить значение яркости. Обозначим этот уровень яркости $v(x, y)$. В случае билинейной интерполяции он задается соотношением

$$v(x, y) \approx a_{00} + a_{10}x + a_{01}y + a_{11}xy \quad (1.3)$$

где коэффициенты $a_{00} - a_{11}$ находятся из системы четырех линейных уравнений с четырьмя неизвестными, выписанной для четырех ближайших соседей точки (x, y) :

$$\begin{bmatrix} a_{00} \\ a_{10} \\ a_{01} \\ a_{11} \end{bmatrix} = \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} x_2 y_2 & -x_2 y_1 & -x_1 y_2 & x_1 y_1 \\ -y_2 & y_1 & y_2 & -y_1 \\ -x_2 & x_2 & x_1 & -x_1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} f(x_1, y_1) \\ f(x_1, y_2) \\ f(x_2, y_1) \\ f(x_2, y_2) \end{bmatrix} \quad (1.4)$$



Ход работы

1. Получить вариант у преподавателя.
2. В соответствии с вариантом, загрузит нужное изображение из папки *test_images* с помощью команды *imread*.
3. **Задание на вращение изображения.**
 - a) Повернутое изображение должно быть такого же размера как исходное. Пиксели выходящие за пределы повернутого изображения нужно обрезать.
 - b) Выполнить сдвиг центра изображения в начало координат.
 - c) Создать пустой массив для нового изображения.
 - d) Выполнить расчет яркости всех пикселей нового изображения, путем поворота пикселей исходного изображения (см. формулу 1.2).
 - e) Выполнить сдвиг центра изображения в центр координат.
 - f) Отобразить изображение с помощью команд *imshow*.
4. **Задание на изменение размера изображения методом билинейной интерполяции:**
 - a) Интерполяция выполняется в два этапа: по строкам и по столбцам. Порядок значения не имеет.
 - b) Увеличить размер исходного изображения в 2 раза путем добавления пустых строк (2,4,6,...).
 - c) Рассчитать коэффициенты a_{00} — a_{11} и значения всех пикселей для добавленной строки (например 2) по формулам (1.3 и 1.4).
 - d) Повторить шаги b-c для столбцов.
 - e) Отобразить изображение с помощью команд *imshow*.