Министерство образования Республики Беларусь

Учреждение Образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра электронных вычислительных средств

Лабораторная работа № 2

«Основы интерполяции изображений»

Вариант 2

|  |  |
| --- | --- |
| Проверил: | Выполнил: |
| Рыбенков Е.В. | ст. гр. 850701 |
|  | Филипцов Д. А. |
|  | |

Минск 2021

# Цель работы

Научиться изменять размеры и выполнять поворот цифровых изображений в MATLAB.

# Задание

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.



Рисунок 1 – Исходное изображение

# Ход работы

MATLAB-код:

I = imread('kodim02.png'); % загрузка изображения

figure('Name','Original');

imshow(I); % вывод изображения

sizeI = size(I);

cx = 1; % коэффициент масштабирования по горизонтали

cy = 1; % коэффициент масштабирования по вертикали

teta = 20; % угол поворота

tx = -sizeI(1)/2; % размер сдвига по горизонтали

ty = -sizeI(2)/2; % размер сдвига по вертикали

sv = 0; % коэффициент вертикального скоса

sh = 0; % коэффициент горизонтального скоса

Ip = preobr(I, cx, cy, teta, tx, ty, sv, sh);

figure('Name','Preobr');

imshow(Ip); % вывод изображения

Код функции preobr:

function [Ip] = preobr(I, cx, cy, teta, tx, ty, sv, sh)

%PREOBR Преобразует изображение по заданным параметрам

% Осуществляет масштабирование, поворот, сдвиг, скос по обеим осям

sizeI = size(I);

Tm = eye(3); % матрица масштабирования

Tm(1, 1) = cx;

Tm(2, 2) = cy;

Tp = eye(3); % матрица поворота

Tp(1, 1) = cos(teta\*pi/180);

Tp(1, 2) = sin(teta\*pi/180);

Tp(2, 1) = -sin(teta\*pi/180);

Tp(2, 2) = cos(teta\*pi/180);

Ts = eye(3); % матрица сдвига

Ts(3, 1) = tx;

Ts(3, 2) = ty;

Ts1 = eye(3); % матрица обратного сдвига

Ts1(3, 1) = -tx;

Ts1(3, 2) = -ty;

Tv = eye(3); % матрица вертикального скоса

Tv(2, 1) = sv;

Tg = eye(3); % матрица горизонтального скоса

Tg(1, 2) = sh;

T = Tm \* Ts \* Tp \* Ts1 \* Tv \* Tg; % матрица итогового преобразования

Ip = zeros(sizeI, 'uint8'); % преобразованная матрица изображения

for x = 1:sizeI(1)

for y = 1:sizeI(2)

p = [x y 1] / T;

p1 = round(p(1));

p2 = round(p(2));

if p1<=sizeI(1) && p2<=sizeI(2) && p1>0 && p2>0

Ip(x, y, :) = I(p1, p2, :);

end

end

end

end

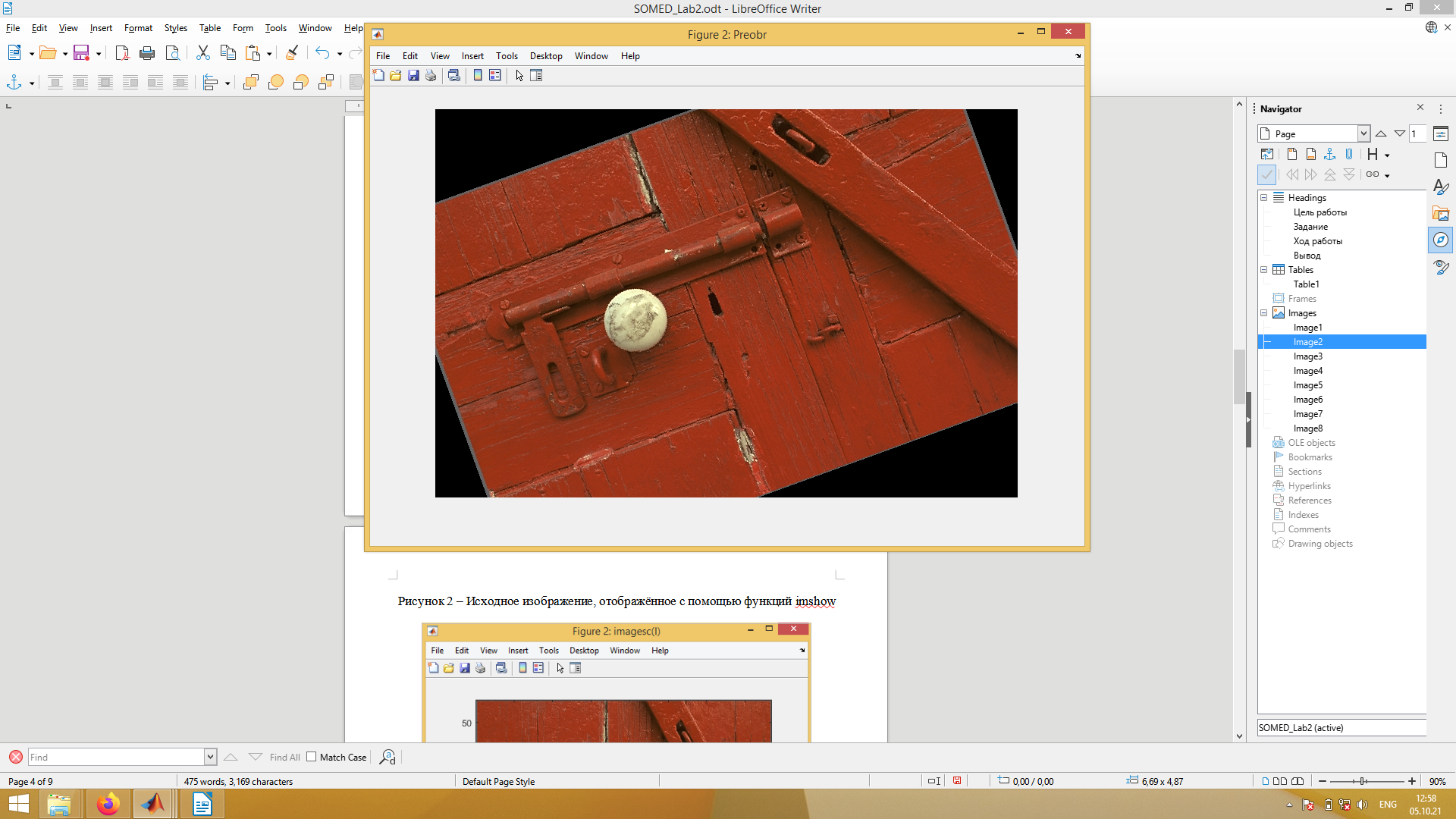


Рисунок 2 – Изображение, повёрнутое на 20 градусов относительно центра

a)Интерполяция выполняет в два этапа: по строком и по столбцам. Порядокзначения не имеет.

b)Увеличить размер исходного изображения 2 два раза путем добавления пустых строк (2,4,6,...).

c)Рассчитать коэффициенты 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.

Код программы:

%% Выполнение интерполяции

Iint = zeros(sizeI .\* [2 1 1], 'uint8'); % интерполированная матрица изображения

Iint(1:2:end, :, :) = I; % увеличение картинки в 2 раза путём добавления пустых строк

sizeIint = size(Iint);

Iint = im2double(Iint);

for y = 2:2:sizeIint(1)

for x = 1:sizeIint(2)

x1 = x - 1;

x2 = x + 1;

y1 = y - 1;

y2 = y + 1;

if x1>0 && x1<=sizeIint(2) && x2>0 && x2<=sizeIint(2) && y1>0 && y1<=sizeIint(1) && y2>0 && y2<=sizeIint(1)

A = (1/((x2 - x1) \* (y2 - y1))) \* [x2\*y2, -x2\*y1, -x1\*y2, x1\*y1; -y2, y1, y2, -y1; -x2, x2, x1, -x1; 1, -1, -1, 1] \* [Iint(y1, x1, 1); Iint(y2, x1, 1); Iint(y1, x2, 1); Iint(y2, x2, 1)];

Iint(y, x, 1) = A(1) + A(2)\*x + A(3)\*y + A(4)\*x\*y;

A = (1/((x2 - x1) \* (y2 - y1))) \* [x2\*y2, -x2\*y1, -x1\*y2, x1\*y1; -y2, y1, y2, -y1; -x2, x2, x1, -x1; 1, -1, -1, 1] \* [Iint(y1, x1, 2); Iint(y2, x1, 2); Iint(y1, x2, 2); Iint(y2, x2, 2)];

Iint(y, x, 2) = A(1) + A(2)\*x + A(3)\*y + A(4)\*x\*y;

A = (1/((x2 - x1) \* (y2 - y1))) \* [x2\*y2, -x2\*y1, -x1\*y2, x1\*y1; -y2, y1, y2, -y1; -x2, x2, x1, -x1; 1, -1, -1, 1] \* [Iint(y1, x1, 3); Iint(y2, x1, 3); Iint(y1, x2, 3); Iint(y2, x2, 3)];

Iint(y, x, 3) = A(1) + A(2)\*x + A(3)\*y + A(4)\*x\*y;

end

end

end

Iint = im2uint8(Iint);

A = [];

Iint1 = zeros(sizeI .\* [2 2 1], 'uint8'); % интерполированная матрица изображения

Iint1(:, 1:2:end, :) = Iint; % увеличение картинки в 2 раза путём добавления пустых столбцов

sizeIint = size(Iint1);

Iint1 = im2double(Iint1);

for y = 1:sizeIint(1)

for x = 2:2:sizeIint(2)

x1 = x - 1;

x2 = x + 1;

y1 = y - 1;

y2 = y + 1;

if x1>0 && x1<=sizeIint(2) && x2>0 && x2<=sizeIint(2) && y1>0 && y1<=sizeIint(1) && y2>0 && y2<=sizeIint(1)

A = (1/((x2 - x1) \* (y2 - y1))) \* [x2\*y2, -x2\*y1, -x1\*y2, x1\*y1; -y2, y1, y2, -y1; -x2, x2, x1, -x1; 1, -1, -1, 1] \* [Iint1(y1, x1, 1); Iint1(y2, x1, 1); Iint1(y1, x2, 1); Iint1(y2, x2, 1)];

Iint1(y, x, 1) = A(1) + A(2)\*x + A(3)\*y + A(4)\*x\*y;

A = (1/((x2 - x1) \* (y2 - y1))) \* [x2\*y2, -x2\*y1, -x1\*y2, x1\*y1; -y2, y1, y2, -y1; -x2, x2, x1, -x1; 1, -1, -1, 1] \* [Iint1(y1, x1, 2); Iint1(y2, x1, 2); Iint1(y1, x2, 2); Iint1(y2, x2, 2)];

Iint1(y, x, 2) = A(1) + A(2)\*x + A(3)\*y + A(4)\*x\*y;

A = (1/((x2 - x1) \* (y2 - y1))) \* [x2\*y2, -x2\*y1, -x1\*y2, x1\*y1; -y2, y1, y2, -y1; -x2, x2, x1, -x1; 1, -1, -1, 1] \* [Iint1(y1, x1, 3); Iint1(y2, x1, 3); Iint1(y1, x2, 3); Iint1(y2, x2, 3)];

Iint1(y, x, 3) = A(1) + A(2)\*x + A(3)\*y + A(4)\*x\*y;

end

end

end

Iint1 = im2uint8(Iint1);

figure('Name','Interp');

imshow(Iint1); % вывод изображения

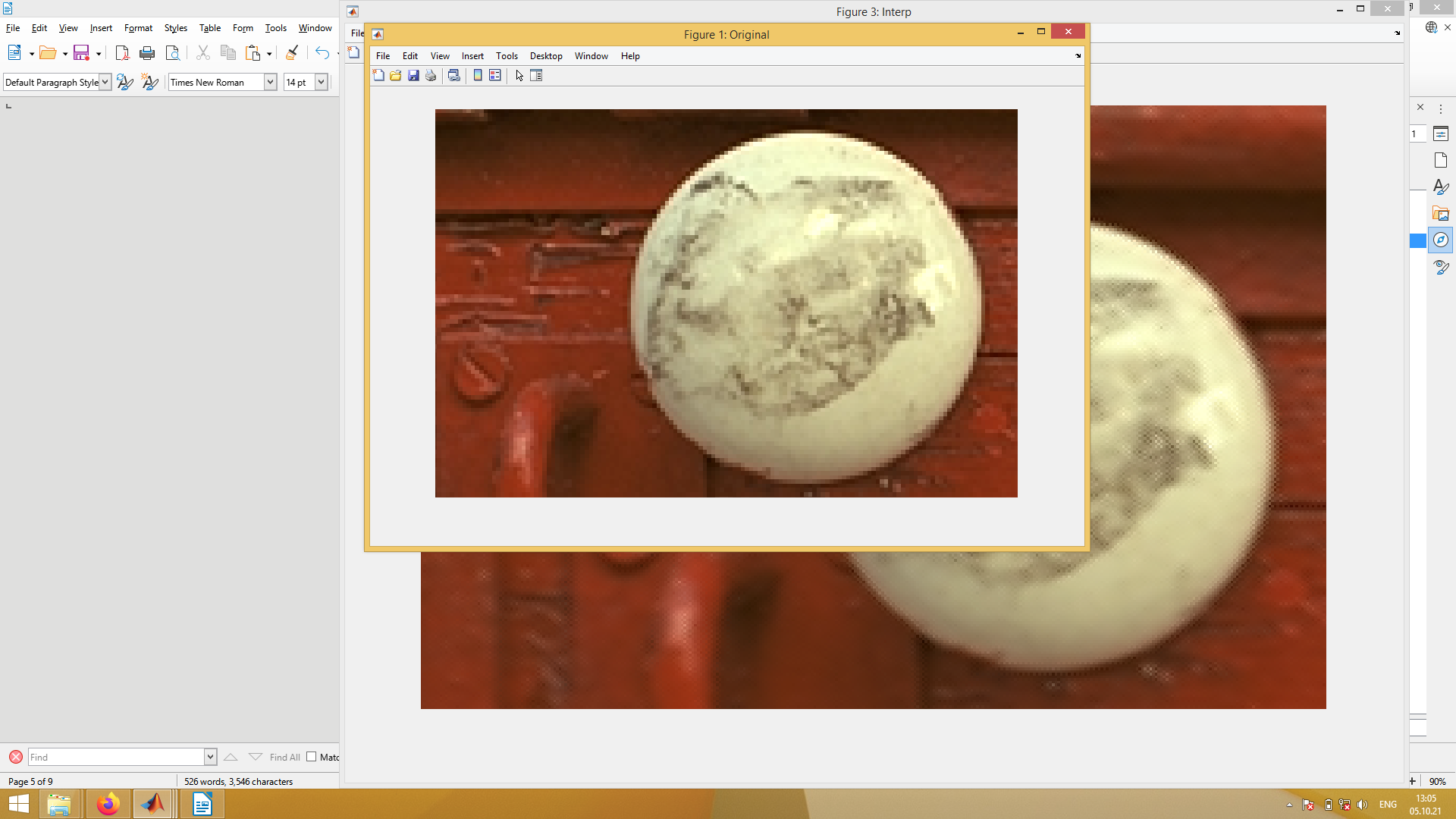


Рисунок 3 – Исходное изображение при увеличении

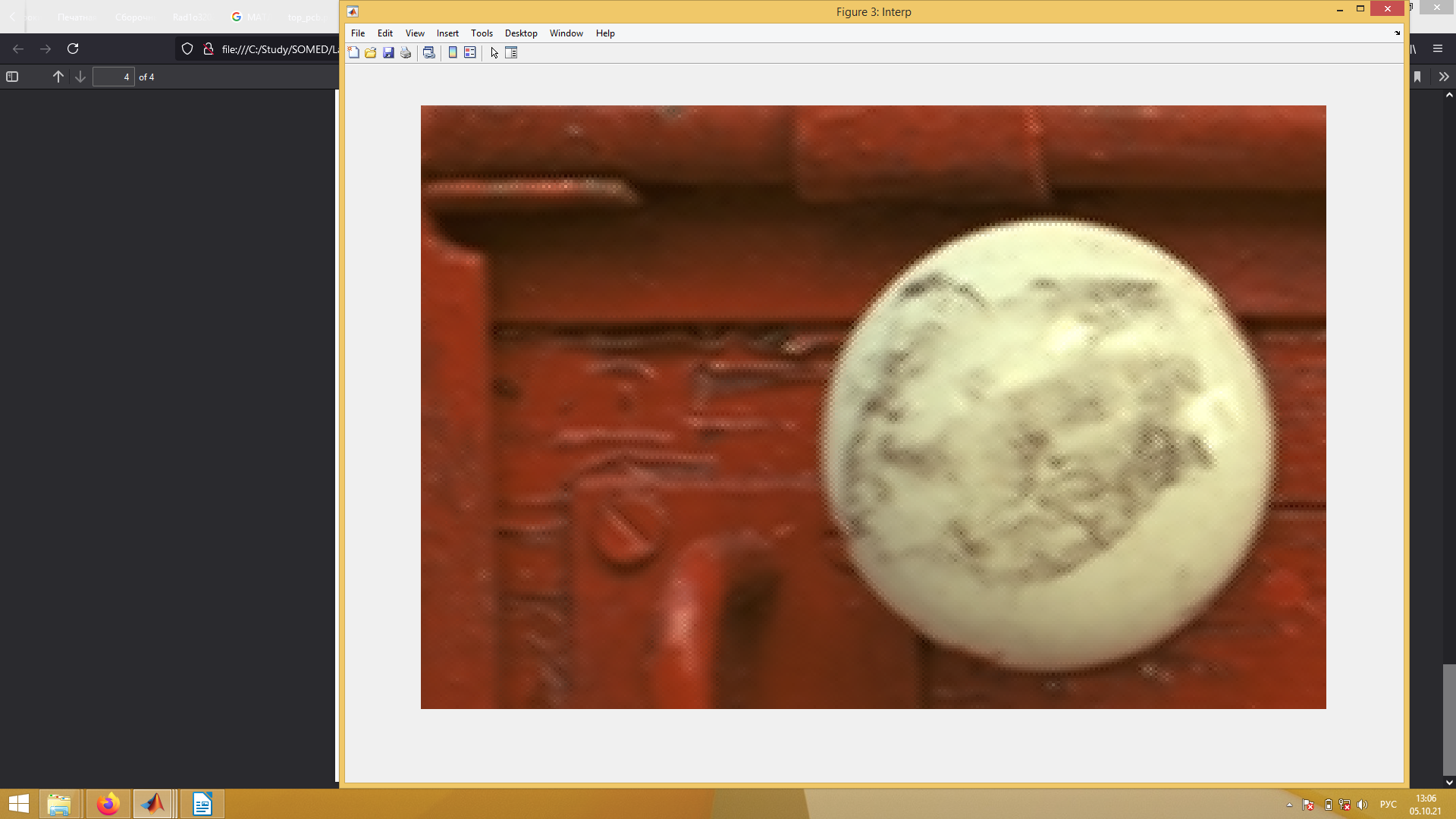


Рисунок 4 – Изображение при увеличении методом билинейной интерполяции

# Вывод

В ходе лабораторной работы были произведены поворот цифрового изображения и увеличение методом билинейной интерполяции, подробно изучены данные алгоритмы.