



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

عنوان: تکلیف اول درس مبانی بینایی کامپیوتر

نام و نام خانوادگی: علیرضا ابره فروش

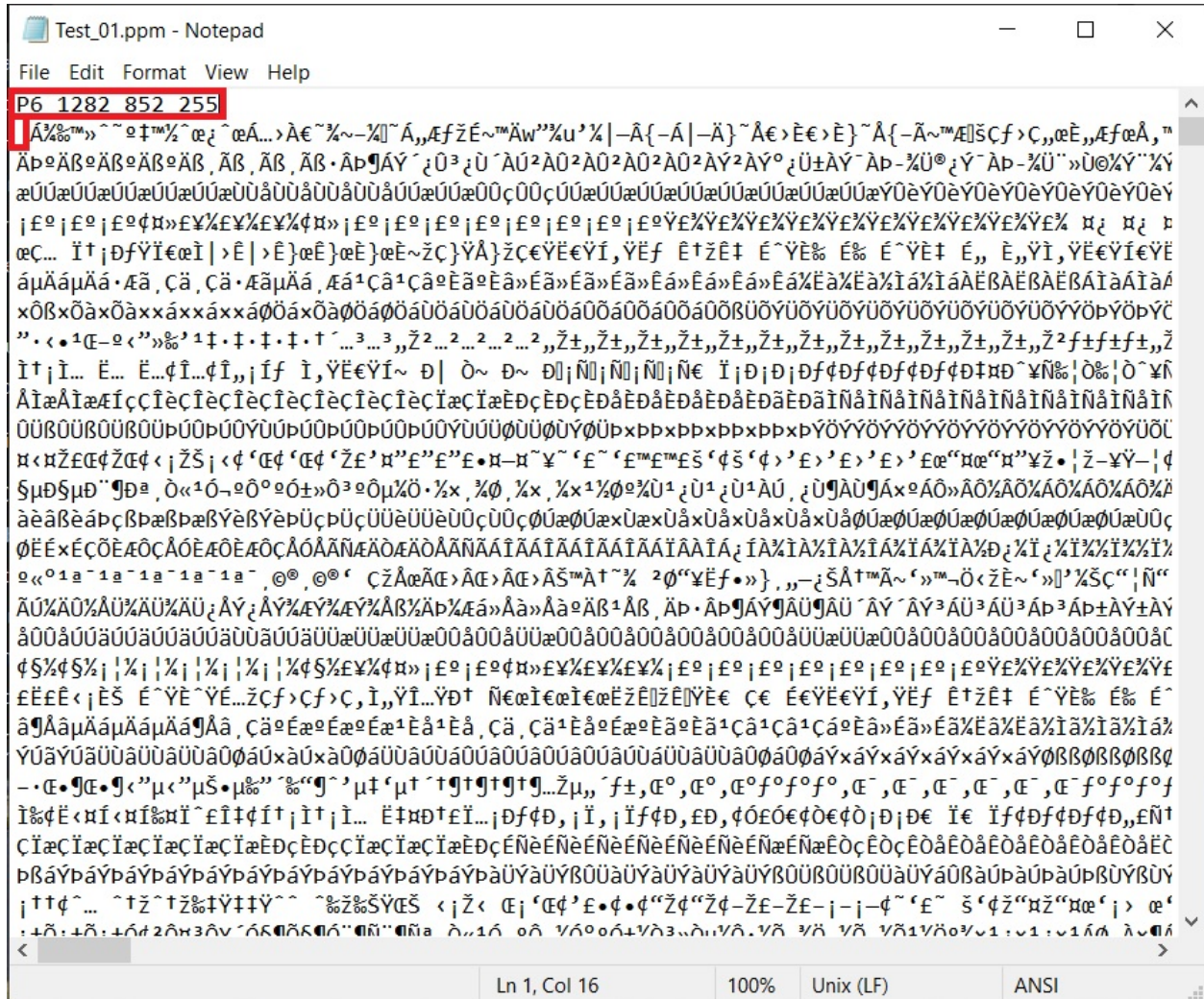
شماره دانشجویی: ۹۸۱۶۶۰۳

نیم سال تحصیلی: بهار ۱۴۰۱/۱۴۰۰

مدرس: دکتر نادر کریمی

دستیاران آموزشی: بهنام ساعدی - محمدرضا مزروعی

ابتدا فایل تصویر را با نرم افزار Notepad باز می کنیم. با توجه به شکل زیر، ابعاد تصویر ۸۵۲×۱۲۸۲ است و همچنین ماکسیمم سطح روشنایی هر کانال رنگی هر پیکسل ۲۵۵ است. در نهایت می توان حدس زد که دیتای تصویر از کاراکتر هفدهم به بعد باشد.



شکل ۱: تصویر

حال تصویر را هم با دستور fopen و fread و هم با استفاده از دستور imtool باز می کنیم. اگر به تصاویر زیر دقت کنیم می بینیم که حدس ما برای اولین کاراکتر دیتای تصویر درست است و سطح روشنایی اولین پیکسل تصویر در بایت هفدهم این فایل قرار دارد. همچنین با مشاهده خروجی دستور imtool و مقایسه با آرایه‌ی نظیر فایل می بینیم که سطح روشنایی نظیر رنگ‌های قرمز، سبز و آبی به ترتیب در اندیس‌های مضرب ۳ به علاوه‌ی ۲، مضرب ۳ و مضرب ۳ به علاوه‌ی ۱ ذخیره شده‌اند.

R:144 G:160 B:193	R:141 G:157 B:190	R:137 G:153 B:187	R:136 G:152 B:186	R:135 G:153 B:189	R:136 G:156 B:191	R:136 G:156 B:193	R:133 G:155 B:192	R:128 G:152 B:190	R:126 G:150 B:188	R:126 G:150 B:188
R:138 G:154 B:187	R:137 G:153 B:186	R:136 G:152 B:186	R:138 G:154 B:188	R:138 G:156 B:192	R:140 G:158 B:194	R:138 G:158 B:195	R:135 G:157 B:194	R:140 G:162 B:201	R:134 G:158 B:196	R:134 G:158 B:196
R:137 G:153 B:187	R:137 G:153 B:187	R:138 G:154 B:190	R:140 G:156 B:192	R:140 G:158 B:194	R:141 G:159 B:195	R:138 G:158 B:195	R:133 G:155 B:192	R:128 G:150 B:189	R:135 G:157 B:196	R:135 G:157 B:196
R:142 G:158 B:194	R:141 G:157 B:193	R:140 G:156 B:192	R:141 G:157 B:193	R:140 G:158 B:194	R:138 G:156 B:192	R:135 G:153 B:191	R:131 G:151 B:188	R:131 G:150 B:190	R:135 G:154 B:194	R:135 G:154 B:194
R:145 G:160 B:199	R:143 G:158 B:197	R:141 G:156 B:195	R:140 G:155 B:194	R:140 G:155 B:194	R:140 G:155 B:194	R:138 G:153 B:192	R:134 G:152 B:190	R:160 G:178 B:216	R:147 G:165 B:203	R:147 G:165 B:203
R:142 G:157 B:196	R:140 G:155 B:194	R:139 G:154 B:193	R:139 G:154 B:193	R:141 G:156 B:195	R:142 G:157 B:196	R:141 G:156 B:195	R:140 G:155 B:194	R:130 G:145 B:184	R:135 G:150 B:189	R:135 G:150 B:189

Pixel info: (10, 1) [126 150 188]

شکل ۲: تصویر

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	32																				
2	54																				
3	32																				
4	49																				
5	50																				
6	56																				
7	50																				
8	32																				
9	56																				
10	53																				
11	50																				
12	32																				
13	50																				
14	53																				
15	53																				
16	10																				
17	144																				
18	160																				
19	193																				
20	141																				
21	157																				
22	180																				
23	137																				
24	153																				
25	187																				
26	136																				
27	152																				
28	186																				
29	135																				
30	153																				

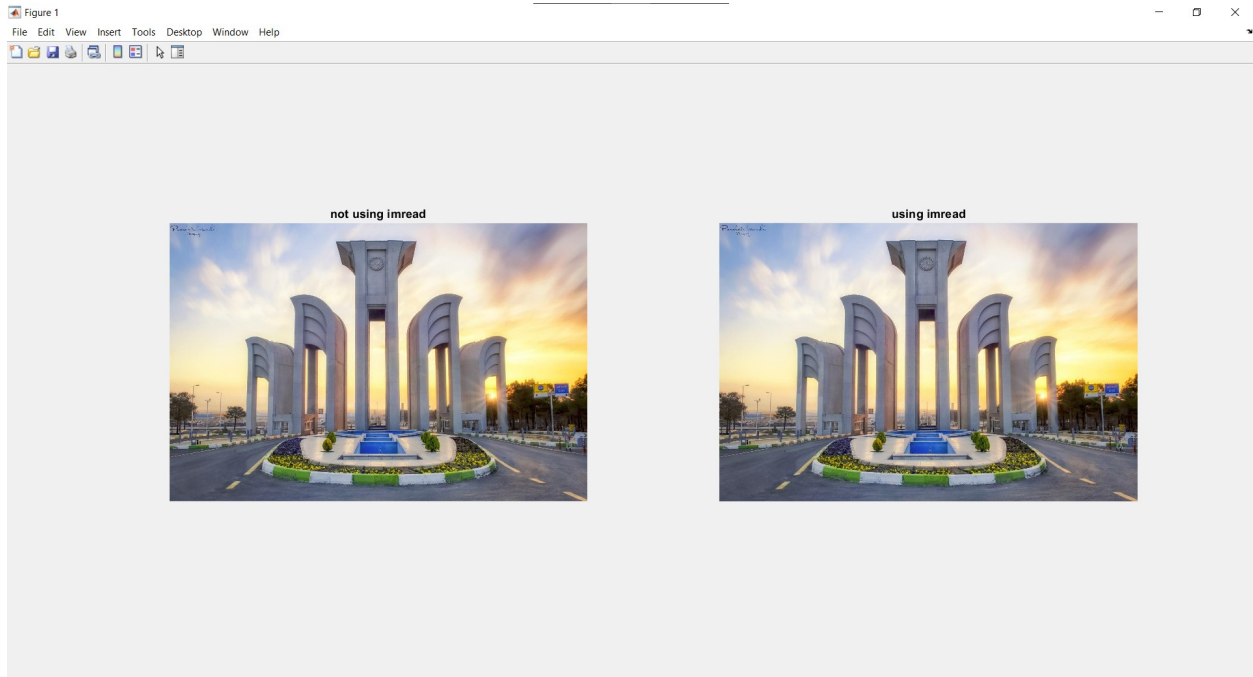
شکل ۳: تصویر

پس برای خواندن تصویر ورودی در محیط برنامه‌نویسی مورد نظر و دسترسی به داده‌های آن به شکل زیر عمل می‌کنیم.

- 1 `%clearing command window and workspace and closing all open figures`
- 2 `clc`
- 3 `clear`
- 4 `close all`

```
5  imtool close all
6  %opening image file and reading its content
7  f = fopen("images/Test_01.ppm");
8  file_content = fread(f, "uint8");
9  %removing ppm header from file_content
10 image_data = file_content(17:end);
11 %defining row and column
12 row = 852;
13 column = 1282;
14 %extracting rgb from image_data
15 R1D = image_data(1:3:end);
16 G1D = image_data(2:3:end);
17 B1D = image_data(3:3:end);
18 %converting 1D rgb to 2D rgb
19 R2D = uint8(zeros(row, column));
20 G2D = uint8(zeros(row, column));
21 B2D = uint8(zeros(row, column));
22 k = 1;
23 for i = 1: row
24     for j = 1: column
25         R2D(i, j) = R1D(k);
26         G2D(i, j) = G1D(k);
27         B2D(i, j) = B1D(k);
28         k = k + 1;
29     end
30 end
31 %concatenating rgb to create the original image
32 I = cat(3, R2D, G2D, B2D);
33 J = imread("images\Test_01.ppm");
34 subplot(1, 2, 1);
35 imshow(I, []);
36 title("not using imread");
37 subplot(1, 2, 2);
38 imshow(J, []);
39 title("using imread");
40 fclose(f);
```

و در آخر می بینیم که خروجی ها یکسان هستند.



شکل ۴: تصویر خروجی سوال ۱

۲

برای این کار می توان ابتدا تصویر را Grayscale کرد، سپس ناحیه ی بیضی شکل را قرمز کرد و در نهایت ناحیه لوزی شکل را با مقادیر اولیه ی تصویر مقداردهی کرد.

```

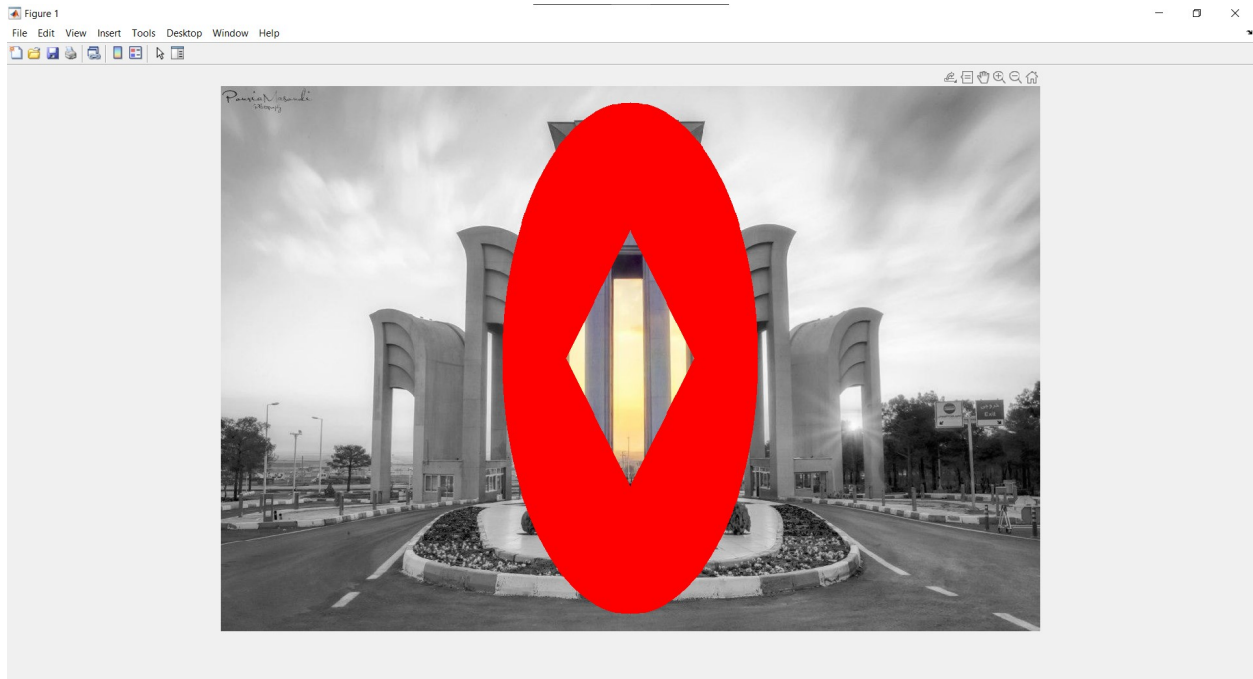
1 %clearing command window and workspace and closing all open figures
2 clc
3 clear
4 close all
5 imtool close all
6 %reading image data
7 I = imread("images\Test_01.ppm");
8 %defining variables
9 row = size(I, 1);
10 column = size(I, 2);
11 ellipse_height = 800;
12 ellipse_width = 400;
13 rhombus_height = 400;
14 rhombus_width = 200;

```

```

15 %generating grayscale image
16 J = uint8(zeros(row, column, 3));
17 J(:, :, 1) = rgb2gray(I);
18 J(:, :, 2) = rgb2gray(I);
19 J(:, :, 3) = rgb2gray(I);
20 %making an ellipse at center of image
21 a = ellipse_width / 2;
22 b = ellipse_height / 2;
23 for i = 1: size(I, 1)
24     for j = 1: size(I, 2)
25         x = column / 2 - j;
26         y = row / 2 - i;
27         if (x ^ 2) / (a ^ 2) + (y ^ 2) / (b ^ 2) <= 1
28             J(i, j, 1) = 255;
29             J(i, j, 2) = 0;
30             J(i, j, 3) = 0;
31         end
32     end
33 end
34 %making rhombus at the center of image
35 for i = 1: size(I, 1)
36     for j = 1: size(I, 2)
37         if (round(abs(i - size(I, 1) / 2)) <= rhombus_width) && (round(abs(j - size(I, 2)
/ 2)) <= rhombus_height) && (round(abs(i - size(I, 1) / 2) * (rhombus_width / 2) +
abs(j - size(I, 2) / 2) * (rhombus_height / 2)) <= rhombus_width * rhombus_height /
4)
38             J(i, j, 1) = I(i, j, 1);
39             J(i, j, 2) = I(i, j, 2);
40             J(i, j, 3) = I(i, j, 3);
41         end
42     end
43 end
44 figure, imshow(J, []);

```

شکل ۵: تصویر خروجی سوال ۲

۳

ابتدا با استفاده از روابط زیر طول و عرض جدید تصویر را محاسبه می‌کنیم:

$$h_{new} = \lceil |w \times \sin(\theta)| + |h \times \cos(\theta)| \rceil$$

$$w_{new} = \lceil |h \times \sin(\theta)| + |w \times \cos(\theta)| \rceil$$

سپس با استفاده از روابط ماتریس دوران، مختصات پیکسل نظیر پیکسل دوران یافته را پیدا می‌کنیم و مقادیر RGBش را برابر آن پیکسل قرار می‌دهیم. تابع rotateImage این کار را انجام می‌دهد.

```

1 function J = rotateImage(I, theta)
2     if mod(theta, 360) ~= 0
3         %defining variables
4         row = size(I, 1);
5         column = size(I, 2);
6         new_row = ceil(abs(column * sind(theta)) + abs(row * cosd(theta)));
7         new_column = ceil(abs(row * sind(theta)) + abs(column * cosd(theta)));
8         J = uint8(zeros(new_row, new_column, 3));
9         for i = 1: size(J, 1)
10             for j = 1: size(J, 2)
11                 [i_p, j_p] = rotatePixel(row, column, i - round((new_row - row) / 2), j -
                    round((new_column - column) / 2), -1 * theta);

```

```

12         if (i_p < 1 || i_p > row || j_p < 1 || j_p > column)
13             J(i, j, 1) = 0;
14             J(i, j, 2) = 0;
15             J(i, j, 3) = 0;
16         else
17             J(i, j, 1) = I(i_p, j_p, 1);
18             J(i, j, 2) = I(i_p, j_p, 2);
19             J(i, j, 3) = I(i_p, j_p, 3);
20         end
21     end
22 end
23 else
24     J = I;
25 end
26 end
27
28 %defining function to calculate new coordinates
29 function [i_p, j_p] = rotatePixel(row, column, i, j, degree)
30 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
31 %defining variables
32     x0 = 0;
33     y0 = 0;
34     x1 = 0;
35     y1 = 0;
36     x2 = 0;
37     y2 = 0;
38     theta = deg2rad(degree);
39 %converting to cartesian coordinate
40 if i <= row / 2
41     y0 = abs(i - row / 2);
42 else
43     y0 = -1 * abs(i - row / 2);
44 end
45 if j <= column / 2
46     x0 = -1 * abs(j - column / 2);
47 else

```



```

48     x0 = abs(j - column / 2);
49     end
50     %calculating rotated cartesian coordinate
51     x1 = x0 * cos(theta) - y0 * sin(theta);
52     y1 = x0 * sin(theta) + y0 * cos(theta);
53     %converting to pixel coordinate
54     x2 = abs(x1);
55     y2 = abs(y1);
56     if (x1 >= 0)
57         j_p = round(column / 2 + x2);
58     else
59         j_p = round(column / 2 - x2);
60     end
61     if (y1 >= 0)
62         i_p = round(row / 2 - y2);
63     else
64         i_p = round(row / 2 + y2);
65     end
66 end

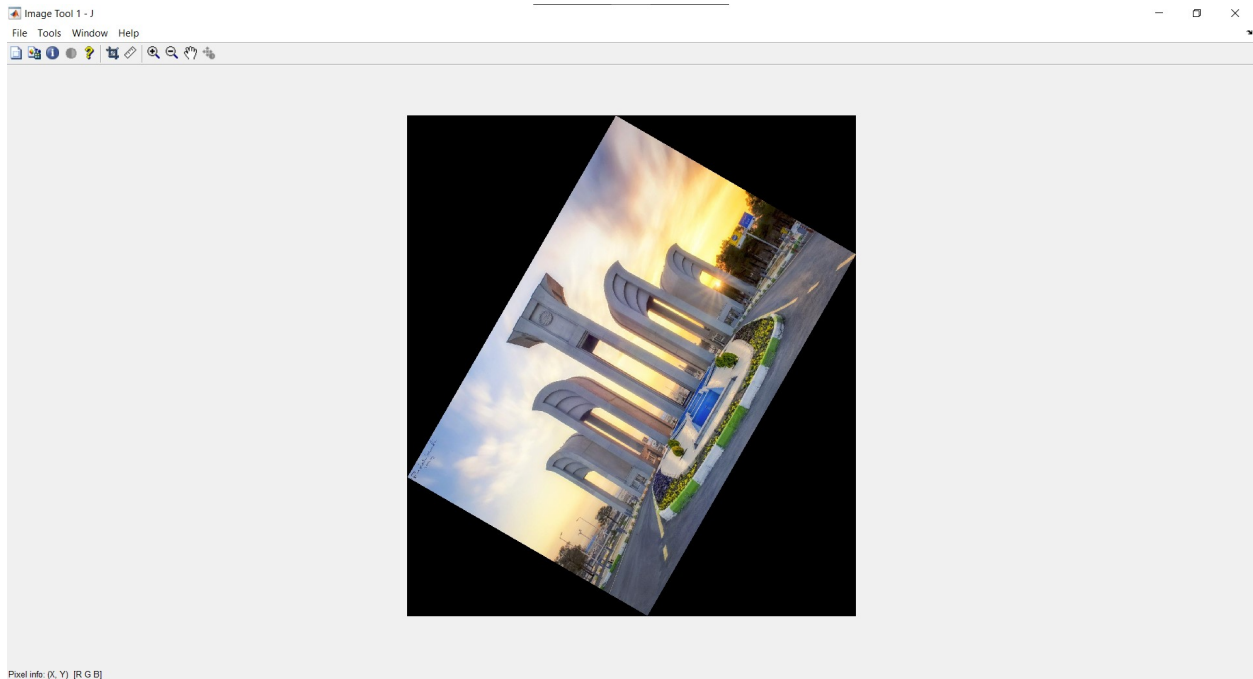
```

در نهایت تابع را به ازای زاویه‌ی ۶۰ درجه فراخوانی می‌کنیم.

```

1  %clearing command window and workspace and closing all open figures
2  clc
3  clear
4  close all
5  imtool close all
6  %reading image data
7  I = imread("images\Test_01.ppm");
8  theta = 60;
9  J = rotateImage(I, theta);
10 figure, imshow(J, []);

```



شکل ۶: نمونه تصویر خروجی سوال ۳ برای زاویه ی ۶۰ درجه

منابع