Part 3: Depth of Field

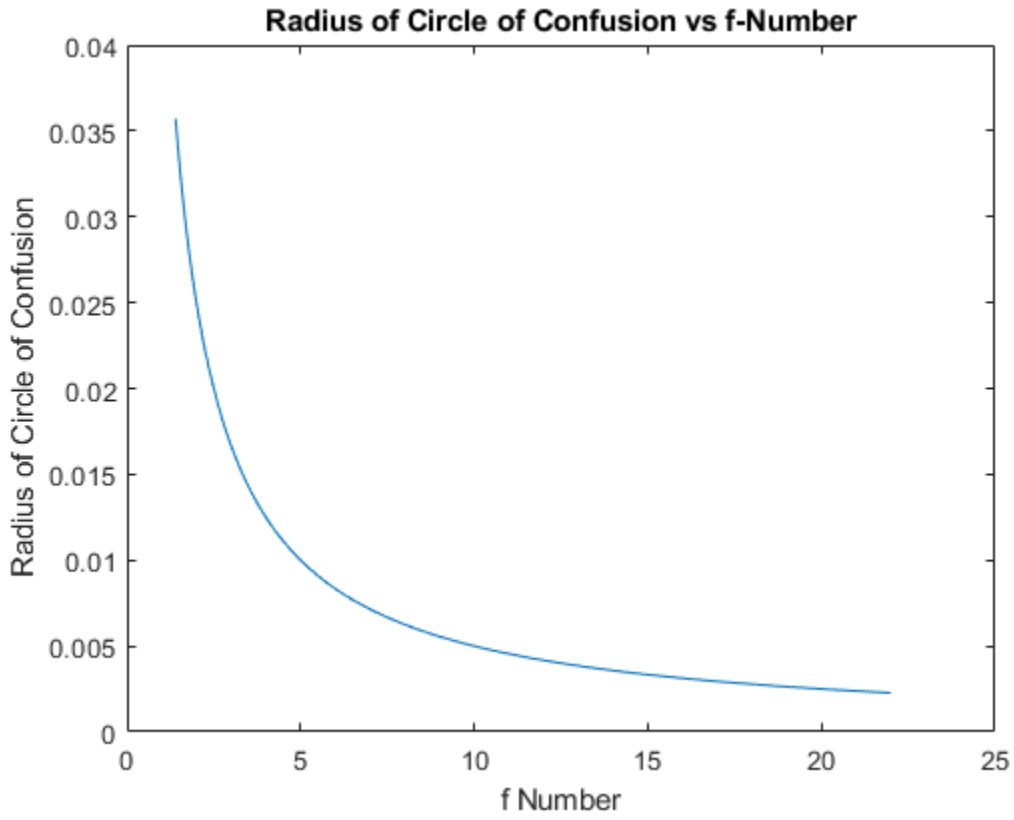
Table of Contents

By: Kaan Dincer and Nick Falshaw

Part 3: Radius of Circle of Confusion vs f-Number

```
f = 0.2;
z1_obj = 4;
z1_lens = 2;
y_in = 0;

fNum = 1.4:.02061:22;

radius = zeros([length(fNum), 1]);
radius = radius';

for i = 1:length(fNum)

    M_z1 = [1, z1_obj;
            0, 1];

    M_f = [1, 0;
        -1/f, 1];

    A = M_f*M_z1;

    z2 = -(z1_lens/(A(2,2)));

    M_z2 = [1, z2;
            0, 1];

    y_height = f / (2*fNum(i));
    theta_in = y_height/z1_obj;

    M_trans = M_z2*M_f*M_z1;

    [y_out, theta_out] = simRayProp(M_trans, y_in, theta_in);

    radius(i) = y_out;
end
```

```
figure();
plot(fNum, radius);
xlabel('f Number');
ylabel('Radius of Circle of Confusion');
title('Radius of Circle of Confusion vs f-Number');
```



Part 3: Image to rays for Turkey

```
z1 = 2; % 2m
f = 0.2; % 200 mm
fNum1 = 22;



M_z1 = [1, z1;
        0, 1];

M_f = [1, 0;
       -1/f, 1];

A = M_f*M_z1;

z2 = -(z1/(A(2,2)));

M_z2 = [1, z2;
```

```
        0, 1];

M_trans1 = M_z2*M_f*M_z1;

turkey = imread('200px-Turkey.png');

w = 0.2; %width
y1 = f/(2*fNum1);
maxAngle = y1/z1;
numRays = 1000000;

[x_out1, y_out1, theta_x_out1, theta_y_out1, color1] =
 img2rays(turkey,w, numRays ,maxAngle);

% simulate rays traveling through the lense
[x_out1, theta_x_out1] = simRayProp(M_trans1, x_out1, theta_x_out1);
[y_out1, theta_y_out1] = simRayProp(M_trans1, y_out1, theta_y_out1);

% Same process with different f-Number:
fNum2 = 1.4;
y2 = f/(2*fNum2);
maxAngle = y2/z1;

[x_out2, y_out2, theta_x_out2, theta_y_out2, color2] =
 img2rays(turkey,w,numRays,maxAngle);

% simulate rays traveling through the lense
[x_out2, theta_x_out2] = simRayProp(M_trans1, x_out2, theta_x_out2);
[y_out2, theta_y_out2] = simRayProp(M_trans1, y_out2, theta_y_out2);
```

Part 3: Images to rays for Snow Globe

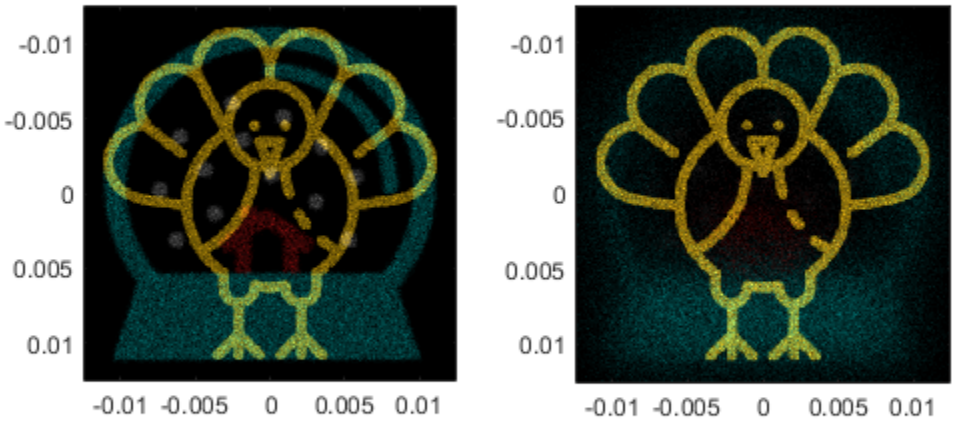
```
z1 = 4; % 4m
f = 0.2; % 200 mm
fNum2 = 1.4;


M_z1 = [1, z1;
        0, 1];

M_f = [1, 0;
       -1/f, 1];

A = M_f*M_z1;

%z2 = -(z1/(A(2,2)));

M_z2 = [1, z2;
        0, 1];

M_trans2 = M_z2*M_f*M_z1;

snowGlobe = imread('200px-Snow_Globe.png');
```

```
w = 0.2; %width
y1 = f/(2*fNum2);
maxAngle = y1/z1;
numRays = 1000000;

[x_out3, y_out3, theta_x_out3, theta_y_out3, color3] =
 img2rays(snowGlobe,w, numRays ,maxAngle);

% simulate rays traveling through the lense
[x_out3, theta_x_out3] = simRayProp(M_trans2, x_out3, theta_x_out3);
[y_out3, theta_y_ou31] = simRayProp(M_trans2, y_out3, theta_y_out3);

% Same process with different f-Number:
fNum2 = 1.4;
y2 = f/(2*fNum1);
maxAngle = y2/z1;

[x_out4, y_out4, theta_x_out4, theta_y_out4, color4] =
 img2rays(snowGlobe,w,numRays,maxAngle);

% simulate rays traveling through the lense
[x_out4, theta_x_out4] = simRayProp(M_trans2, x_out4, theta_x_out4);
[y_out4, theta_y_out4] = simRayProp(M_trans2, y_out4, theta_y_out4);
```

Part 3: Rays to Image

```
rays1_xconcat1 = cat(2,x_out1, x_out3);
rays1_yconcat1 = cat(2,y_out1, y_out3);
rays1_colorConcat1 = cat(2,color1, color3);

rays2_xconcat2 = cat(2,x_out2, x_out4);
rays2_yconcat2 = cat(2,y_out2, y_out4);
rays2_colorConcat2 = cat(2,color2, color4);

Npixels = 500;
sensorWidth = 0.025;

[newimg1, x_1, y_1] = rays2img(rays1_xconcat1, rays1_yconcat1,
 rays1_colorConcat1,sensorWidth,Npixels);

newimg1 = fliplr(newimg1);

[newimg2, x_2, y_2] = rays2img(rays2_xconcat2, rays2_yconcat2,
 rays2_colorConcat2,sensorWidth,Npixels);

newimg2 = fliplr(newimg2);

figure()
subplot(1,2,1);
imagesc([x_2(1);x_2(2)],[y_2(1);y_2(2)],newimg2);

axis image;
```

```
subplot(1,2,2);
imagesc([x_1(1);x_1(2)],[y_1(1);y_1(2)],newimg1);

axis image;
```



Published with MATLAB® R2019a