
Table of Contents

.....	1
Sample Plot	1
Light Field	2
Focus	3
Space Image	5
Computational imaging	6
Lightfield Revisited	7

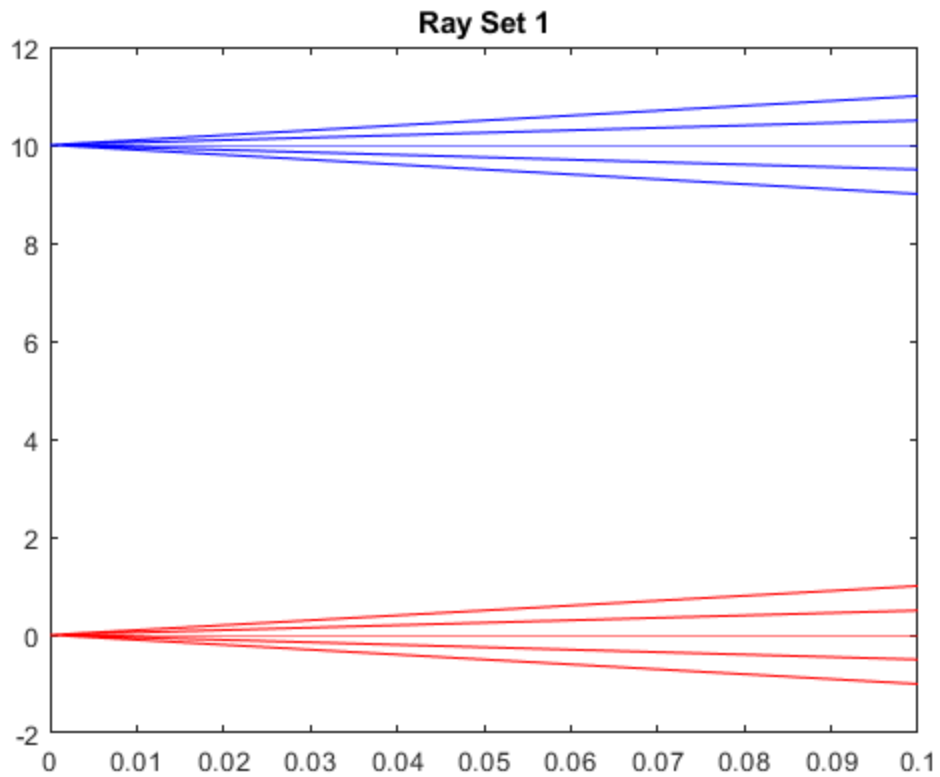
```
load ("lightField.mat");
```

Sample Plot

```
d = 0.1;
M = [1 d 0 0;
     0 1 0 0;
     0 0 1 d;
     0 0 0 1];

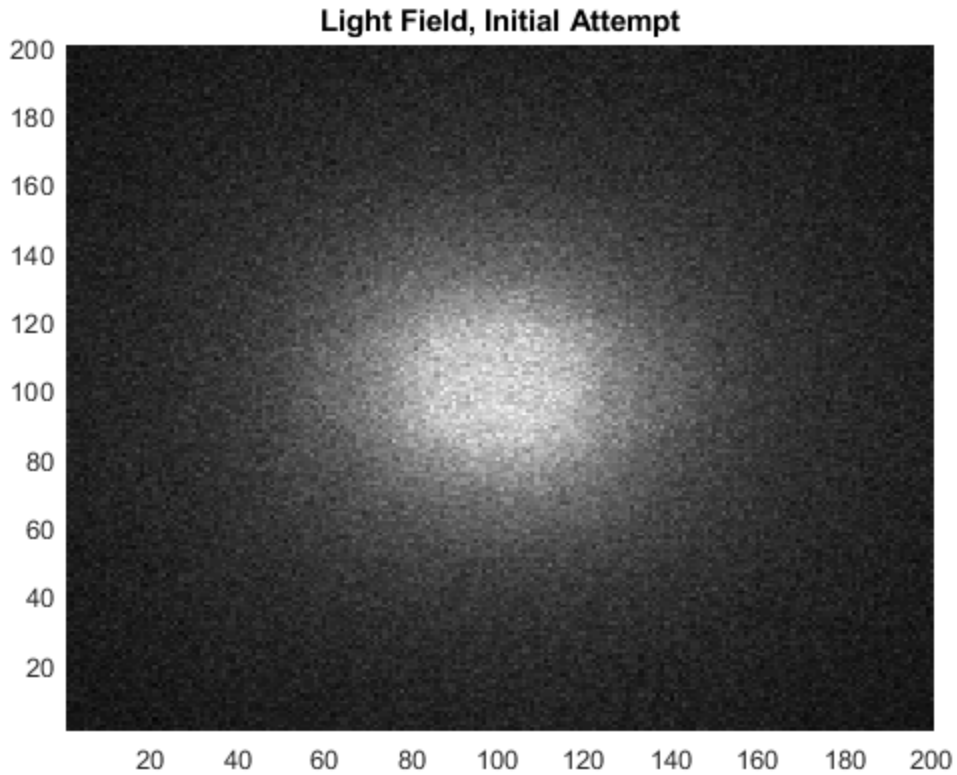
rays_in0 = [0 0 0 0 0;
            10 5 0 -5 -10;
            0 0 0 0 0;
            10 5 0 -5 -10];
rays_out0 = M*rays_in0;
rays_in10 = [10 10 10 10 10;
             10 5 0 -5 -10;
             10 10 10 10 10;
             10 5 0 -5 -10];
rays_out10 = M*rays_in10;
ray_z = [zeros(1,size(rays_in0,2)); d*ones(1,size(rays_in0,2))];

figure(1)
plot(ray_z, [rays_in0(1,:); rays_out0(1,:)], Color="r");
hold on;
plot(ray_z, [rays_in10(1,:); rays_out10(1,:)], Color="b");
title("Ray Set 1")
hold off;
```



Light Field

```
img = rays2img(rays(1,:),rays(3,:),0.008,200);  
figure(2);  
hold on;  
image(img);  
axis tight  
title("Light Field, Initial Attempt")  
colormap ("gray")  
hold off;  
  
% new = M*rays;  
% img = rays2img(new(1,:),new(3,:),0.02,200);  
% figure(3);  
% hold on;  
% image(img);  
% colormap ("gray")  
% hold off;
```



Focus

```
d1 = 0.2;  
d2 = 0.43;  
f = 0.125;  
  
M1= [1 d1 0 0;  
      0 1 0 0;  
      0 0 1 d1;  
      0 0 0 1];  
M2= [1 d2 0 0;  
      0 1 0 0;  
      0 0 1 d2;  
      0 0 0 1];  
Mf= [1 0 0 0;  
      (-1/f) 1 0 0;  
      0 0 1 0;  
      0 0 (-1/f) 1];  
  
rays_in0 = [0 0 0 0 0;  
            2 1 0 -1 -2;  
            0 0 0 0 0;  
            2 1 0 -1 -2];  
rays_out0M1 = M1*rays_in0;
```

```

rays_out0Mf = Mf*rays_out0M1;

rays_out0M2= M2*rays_out0Mf;

x = 0.2;
rays_in1 = [x x x x x;
            2 1 0 -1 -2;
            x x x x x;
            2 1 0 -1 -2];
rays_out1M1 = M1*rays_in1;

rays_out1Mf = Mf*rays_out1M1;

rays_out1M2 = M2*rays_out1Mf;

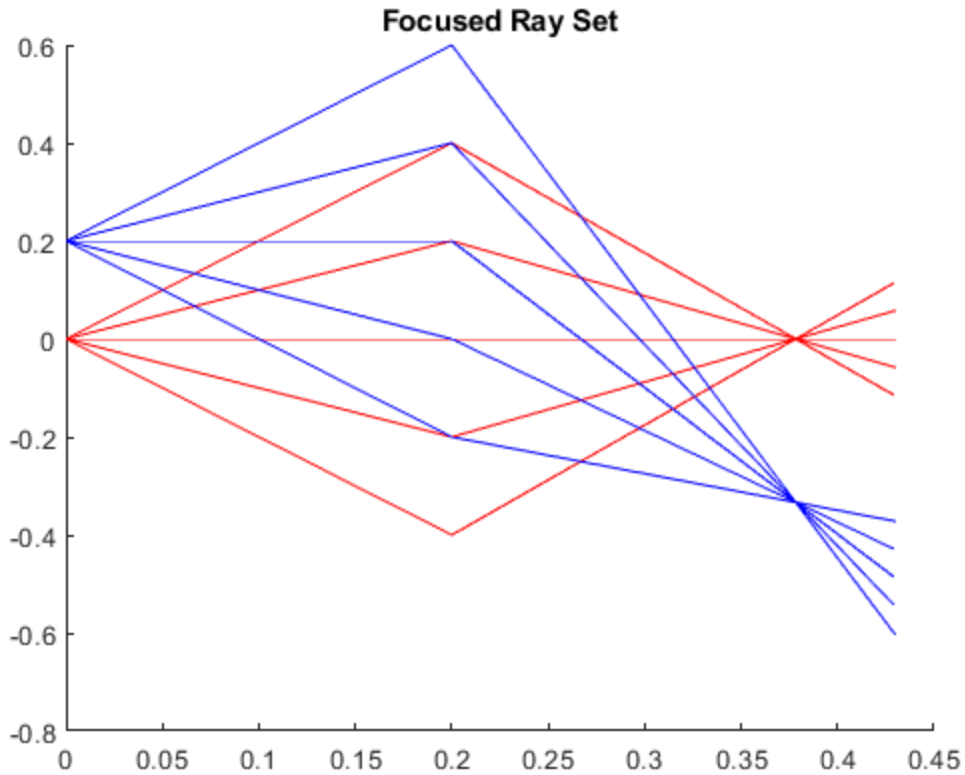
ray_z1 = [zeros(1,size(rays_in0,2)); d1*ones(1,size(rays_in0,2))];
ray_z2 = [d1*ones(1,size(rays_in0,2)); d2*ones(1,size(rays_in0,2))];

figure(4)
hold on;

plot(ray_z1, [rays_in0(1,:); rays_out0M1(1,:)], Color="r");
plot(ray_z1, [rays_in1(1,:); rays_out1M1(1,:)], Color="b");

plot(ray_z2, [rays_out0Mf(1,:); rays_out0M2(1,:)], Color="r");
plot(ray_z2, [rays_out1Mf(1,:); rays_out1M2(1,:)], Color="b");
title("Focused Ray Set")
hold off;

```



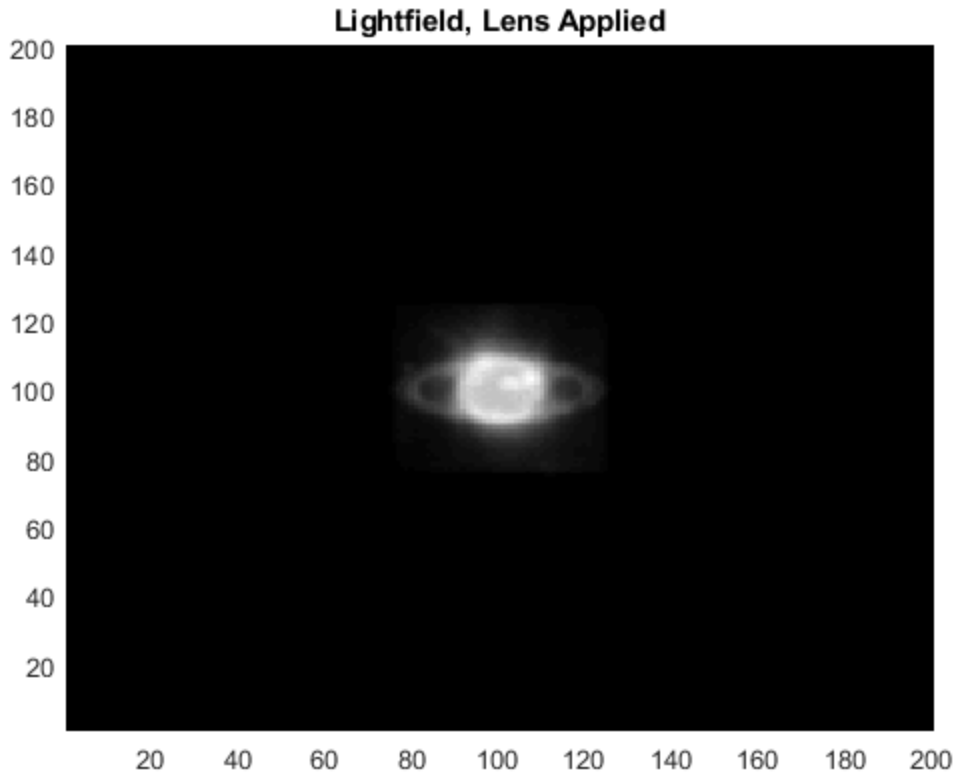
Space Image

```
f = 0.335; % Lens width used to make image below
d2 = 0.5; % Should be same as lens width, not sure why this is best.
```

```
M2= [1 d2 0 0;
      0 1 0 0;
      0 0 1 d2;
      0 0 0 1];
Mf= [1 0 0 0;
      (-1/f) 1 0 0;
      0 0 1 0;
      0 0 (-1/f) 1];

rays_new = M2*Mf*rays;

img = rays2img(rays_new(1,:),rays_new(3,:),0.008,200);
figure(5);
hold on;
image(img);
colormap ("gray")
title("Lightfield, Lens Applied")
axis tight
hold off;
```



Computational imaging

```
%Create an inverted matrix
%I understand that this isn't the correct way to do it
%FIXME replace with actual method

%Start off with an identity matrix that we will use to compare with M
Mi = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];

%Now we need to find the inverse, for that all we have to do is convert M
%to an identity matrix and store the changes into Mi, which will become the
%inverse
%M is a very simple matrix and the only changes we have to do is cancel out
%the d values from it, this means putting -d values into Mi in their
%respective positions

%I will make a copy of M so it isn't damaged
M3 = M;
d = 0.1;

%Now we apply the same methods to both matrices to convert M3 to an
%identity and make Mi an inverse
M3(1,:) = M(1,:) - M(2,:)*d;
Mi(1,:) = Mi(1,:) - Mi(2,:)*d;
```

```
M3(3,:) = M(3,:) - M(4,:)*d;  
Mi(3,:) = Mi(3,:) - M(4,:)*d;
```

```
%Compared to an M matrix where the d values were made negative, the inverse  
%of M and M-d are equivalent to one another
```

Lightfield Revisited

```
%Create a new matrix that is equal to the inverse of M with a selected  
%distance
```

```
%A distance of 1 is perfect in this case  
d = 1;
```

```
%Determine the base matrix and the identity  
M4 = [1 d 0 0; 0 1 0 0; 0 0 1 d; 0 0 0 1];
```

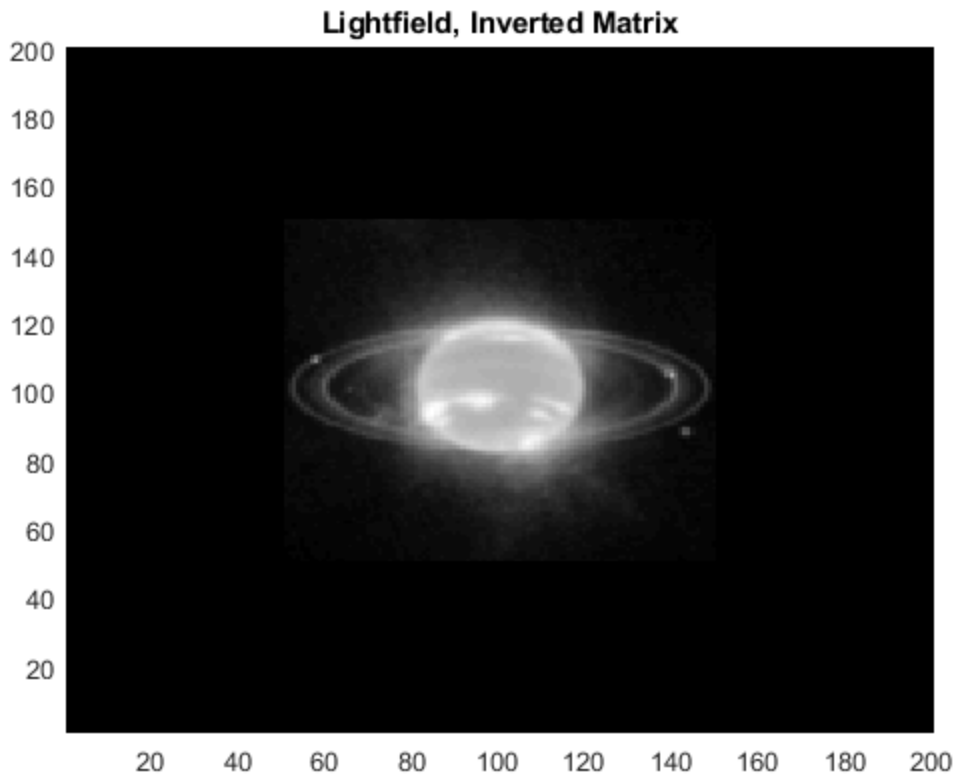
```
M4i = [1 0 0 0; 0 1 0 0; 0 0 1 0; 0 0 0 1];
```

```
%Turn the identity into the inverse  
M4(1,:) = M4(1,:) - M4(2,:)*d;  
M4i(1,:) = M4i(1,:) - M4i(2,:)*d;
```

```
M4(3,:) = M4(3,:) - M4(4,:)*d;  
M4i(3,:) = M4i(3,:) - M4i(4,:)*d;
```

```
%Process the results  
rays_final = M4i*rays;
```

```
img = rays2img(rays_final(1,:),rays_final(3,:),0.008,200);  
figure(6);  
hold on;  
image(img);  
colormap ("gray")  
title("Lightfield, Inverted Matrix")  
axis tight  
hold off;
```



Published with MATLAB® R2022a