
PolarimetryofImagesEG16

Table of Contents

.....	1
Read input	1
Extracting the information	1
Plotting the ellipse orientation	2
Plotting the ellipses	3

Program to do polarimetry of images: to find the state of polarization of each imaged point.

The input consists of 6 files from a digital camera that are taken after the light passes through 6 polarization filters: horizontal, vertical, diagonal, antidiagonal, right-circular, and left-circular. The output is an image file showing in false color the orientation of the polarization ellipse. The saturation of the color encodes the total intensity of the beam. Other parameters within the program are: switchellipse = 1 to not plot ellipses; otherwise plot; centerx = The x-coordinate of the center of the image in pixels; centery = The y-component of the center of the image in pixels; delpix = The width of the image in pixels. The program automatically detects the size of the file in pixels putting them in variables Nx and Ny. If you make centerx=Nx/2, centery=Ny/2 and delpix=Nx, the entire field of the camera is shown and then one can narrow the values of those variables.

Written by Enrique J. Galvez, Colgate University 2012-2016 Adapted from a program by Giovanni Milione 2012 Copyright. Not for commercial use.

```
close all
clear;
```

Read input

```
h = double(imread('H.jpg')); % Horizontal image
a = double(imread('A.jpg')); % Antidiagonal image
v = double(imread('V.jpg')); % Vertical image
d = double(imread('D.jpg')); % Diagonal image
l = double(imread('L.jpg')); % Left circular image
r = double(imread('R.jpg')); % Right circular image
centerx = 1734;
centery = 900;
delpix = 1200;
switchellipse = 0; % if 1 skip ellipses plot, if 0 do ellipses KG 9/19
ellis='4'; %label for the graph of the value of l used
```

Extracting the information

```
[Nx,Ny]=size(h(:,:,1)) %% This determines the size of the image
% This defines the unnormalized Stokes parameters
```

```
MH=h(1:Nx,1:Ny,1); % Matrix with horizontal filter
MV=v(1:Nx,1:Ny,1); % Matrix with vertical filter
MD=d(1:Nx,1:Ny,1); % Matrix with diagonal filter
MA=a(1:Nx,1:Ny,1); % Matrix for antidiagonal filter
MR=r(1:Nx,1:Ny,1); % Matrix for right circular filter
ML=l(1:Nx,1:Ny,1); % Matrix for left circular filter

% Subtracting background in case it is high
MR = MR-min(min(MR));
ML = ML - min(min(ML));
MH = MH-min(min(MH));
MV = MV - min(min(MV));
MD = MD - min(min(MD));
MA = MA- min(min(MA));

%These are the normalized Stokes parameters
S0=(MR+ML+MH+MV+MA+MD)/3; % Averaged total intensity of the beam
maximum=max(max(S0)); % Max value of the intensity
S0=(MR+ML)./maximum; % Normalized Stokes parameters
S1=(MH-MV)./maximum;
S2=(MD-MA)./maximum;
S3=(MR-ML)./maximum;
Snormsq=S1.^2+S2.^2+S3.^2; % Length of Stokes vector in case it is less than 1
Snorm=Snormsq.^(1/2); % Length of Stokes vector in case it is less than 1
Degpol=Snorm./S0; % Degree of polarization
psi=.5*mod(angle(S1+1i*S2),2*pi); % Orientation angle of the ellipse
epsilon=S3./Snorm; %This is the ellipticity
chi=(asin(epsilon)/2); % Half of latitude on the Poincare Sphere
chideg=chi*180/pi; % in degrees
```

Nx =

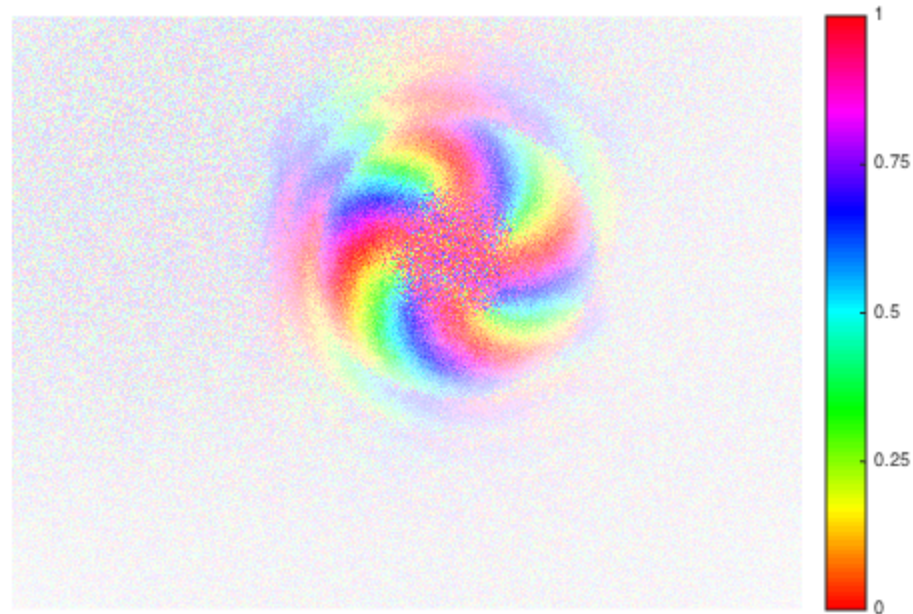
2304

Ny =

3072

Plotting the ellipse orientation

```
h1=figure(1);
clims=[0 pi]; % this fixes the color scale for phase
imshow(psi, 'DisplayRange',clims, 'InitialMagnification', 'fit');
axis image
colormap(hsv(128)); % this option plots the orientation
hcb = colorbar('YTickLabel',{'0','0.25','0.5','0.75','1','1.25','1.5','1.75','2'});
set(hcb, 'YTickMode', 'manual');
set(hcb, 'YTick',[0 pi/4 pi/2 3*pi/4 pi 5*pi/4 3*pi/2 7*pi/4 2*pi]);
alpha(S0);
```



Plotting the ellipses

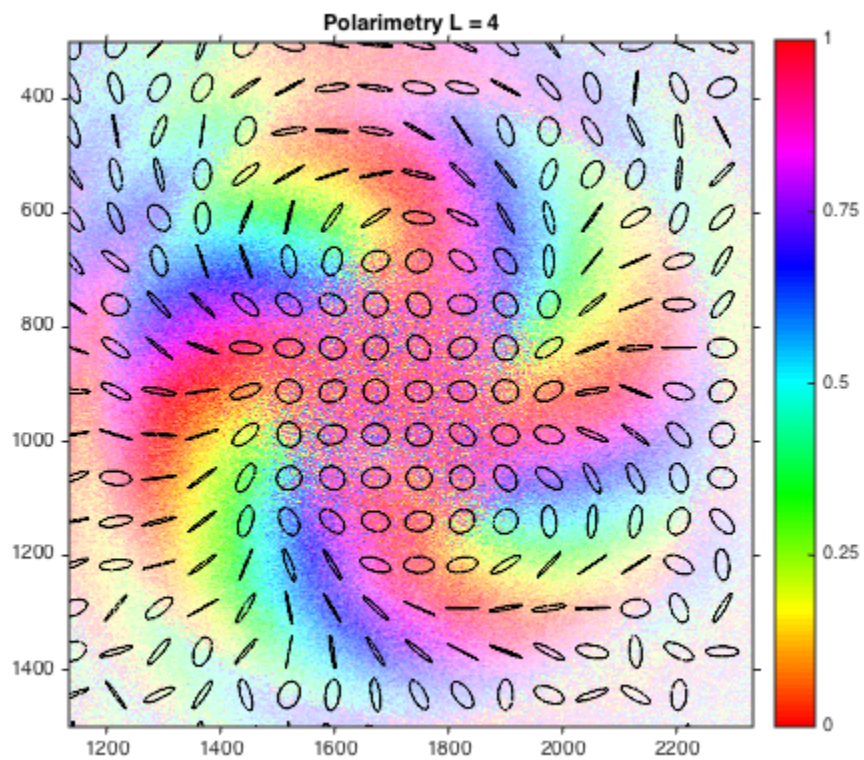
this is a switch so we do not plot the ellipses KG 9/19

```
if switchellipse == 1
else
    N = min(Nx,Ny); % Should be the smaller of width of camera in pixels
    c=30; % size of the ellipse
    step=30; %number of ellipses drawn
    while rem(N,step) ~= 0 %Reduce N to get integer values
        N=N-1;
    end
    mx = N/step; %vertical increment (must be integer)
    my = N/step; %horizontal increment (must be integer)
    Xc=0:pi/500:2*pi;
    hold on
    for i = 1:step;
        for j = 1:step;
            x = mx*i; y = my*j;
            Ex=(cos(chi(x,y)).*cos(Xc)*cos(-psi(x,y))-sin(chi(x,y)).*sin(Xc)*sin(-
            Ey=(cos(chi(x,y)).*cos(Xc)*sin(-psi(x,y))+sin(chi(x,y)).*sin(Xc)*cos(-
            if S0(x,y)<0.1 % to not draw ellipses where intensity is low
            else
                plot(Ex,Ey, 'k', 'LineWidth',1)
            end
        end
    end
end
```

```

end
end
hold off
axis on
axis square
xlo= centerx-delpix/2;
xhi= centerx+delpix/2;
ylo= centery-delpix/2;
yhi= centery+delpix/2;
axis([xlo xhi ylo yhi]);
title(['Polarimetry L = ',ell1s]);
Inam = strcat('PolarimetryL',ell1s, '.bmp');
print(h1, '-dbmp', Inam);

```



Published with MATLAB® R2014b