VISIÓ PER

COMPUTADOR

Informe de laboratori 2

Marc Cervilla Rovira

03/03/2021

Introducció:

En aquest segon exercici de laboratori seguirem practicant amb el Matlab i l’objectiu d’aquesta vegada serà fusionar dues imatges de la nebulosa d’Orion per tal d’aconseguir millor contrast i reduir el soroll. De manera que d’unes imatges originals com la imatge a) a un resultat final com la imatge b).



1. b)

Implementació:

Per aquesta pràctica ens donen el següent guió de treball:

Llegim les imatges com a doubles:

A = double(imread('\_MG\_7735.JPG'))/255;

B = double(imread('\_MG\_7737.JPG')) /255;

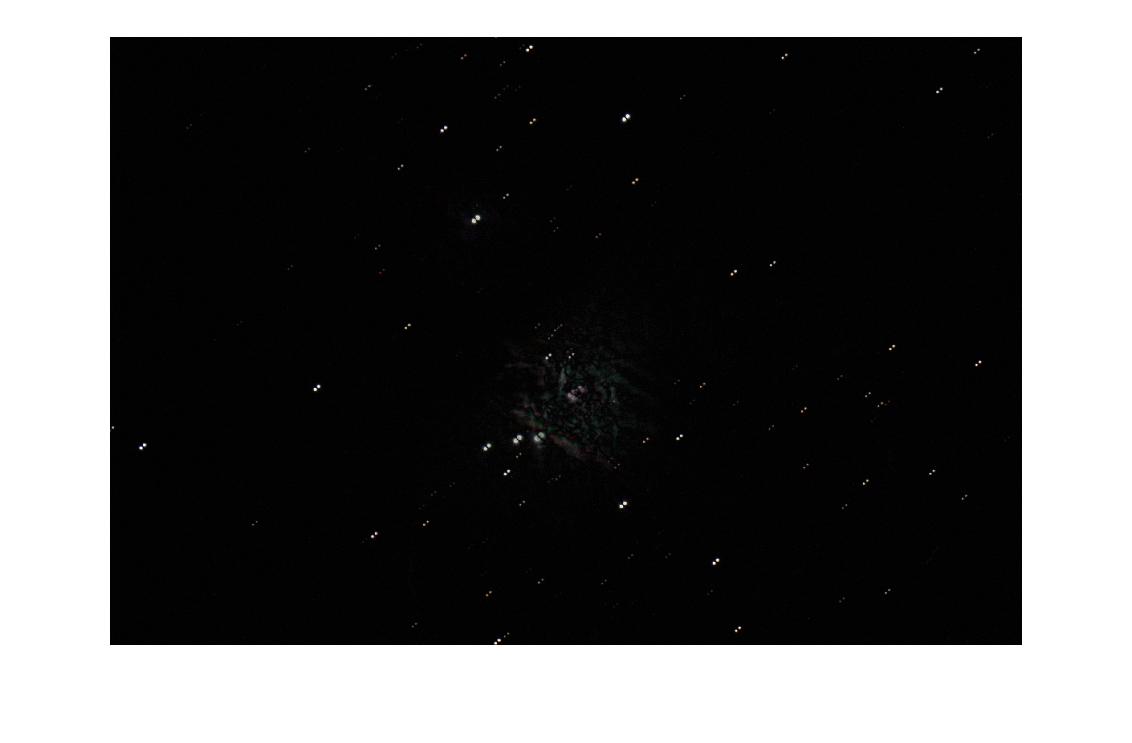
Restem les imatges per veure la seva diferència i les reescalem a valors entre 0 i 1:

DIF = abs(A-B); % imatge diferencia

maxim = max(DIF(:));

DIF = DIF/maxim; % dividim pel seu valor màxim

imshow(DIF);



El que observem es que, degut a la rotació terrestre, les imatges solapades apareixen mogudes, per tant, hem d’apliar una translació per a que aquestes s’adjustin a la mateixa posició:

Bd = imtranslate(B,[20, -20]);

DIF = abs(A-Bd);

maxim = max(DIF(:));

DIF = DIF/maxim;

imshow(DIF);

Aqui podem observar com les imatges ja estan ben ajustades, ja que amb la resta d’imatges les figures de l’imatge desapareixen:



Ara ja podem sumar les imatges per obtenir la imatge millorada i el que hem de fer és millorar la il·luminació i el contrast per obtenir la imatge final:

Am = (A+Bd)/2; % imatge millorada

Am = XXXXXX;

montage ({A,Am});

On fica XXXXXX he anat provant diferents funcions no lineals per tal d’aconseguir l’objectiu.

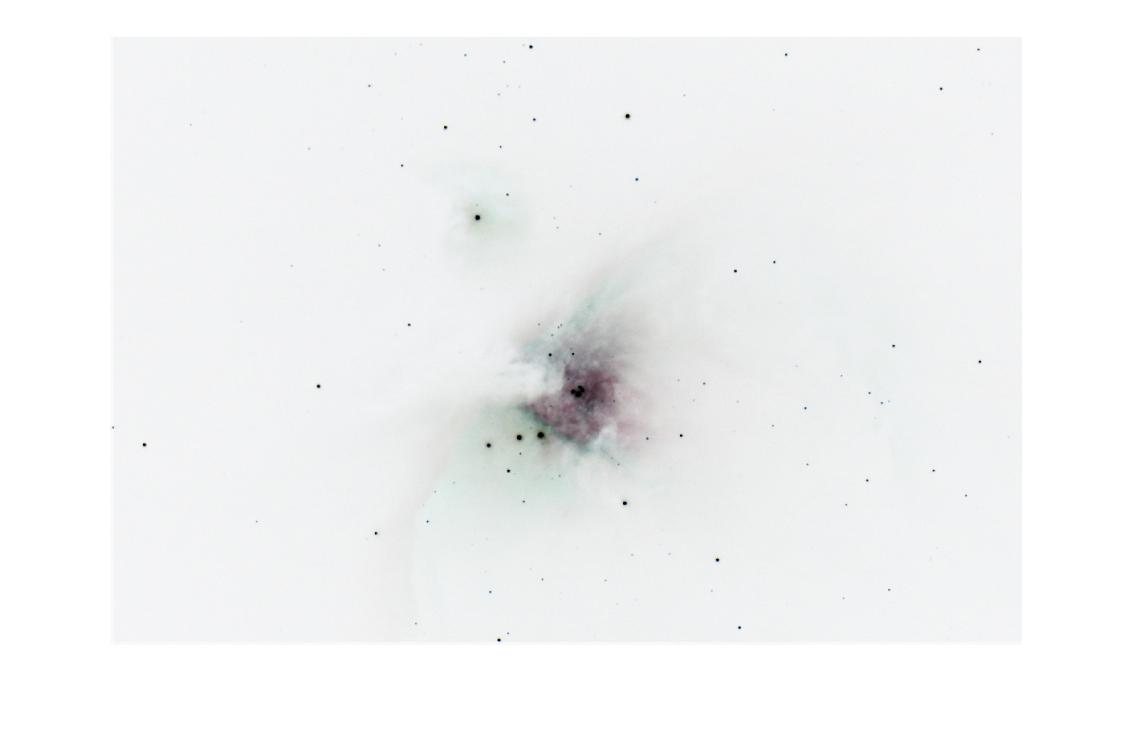
En primer lloc, he provat d’utilitzar la funció arrayfun. Amb això apliquem una funció a cada pixel de l’imatge i he provat diferents funcions com logaritmes o per exemple l’arrel quadrada:

arrayfun(@(x) sqrt(x), Am)

Però l’execució d’aquest codi tarda bastant i els resultats no són el que busquem com veiem a continuació:



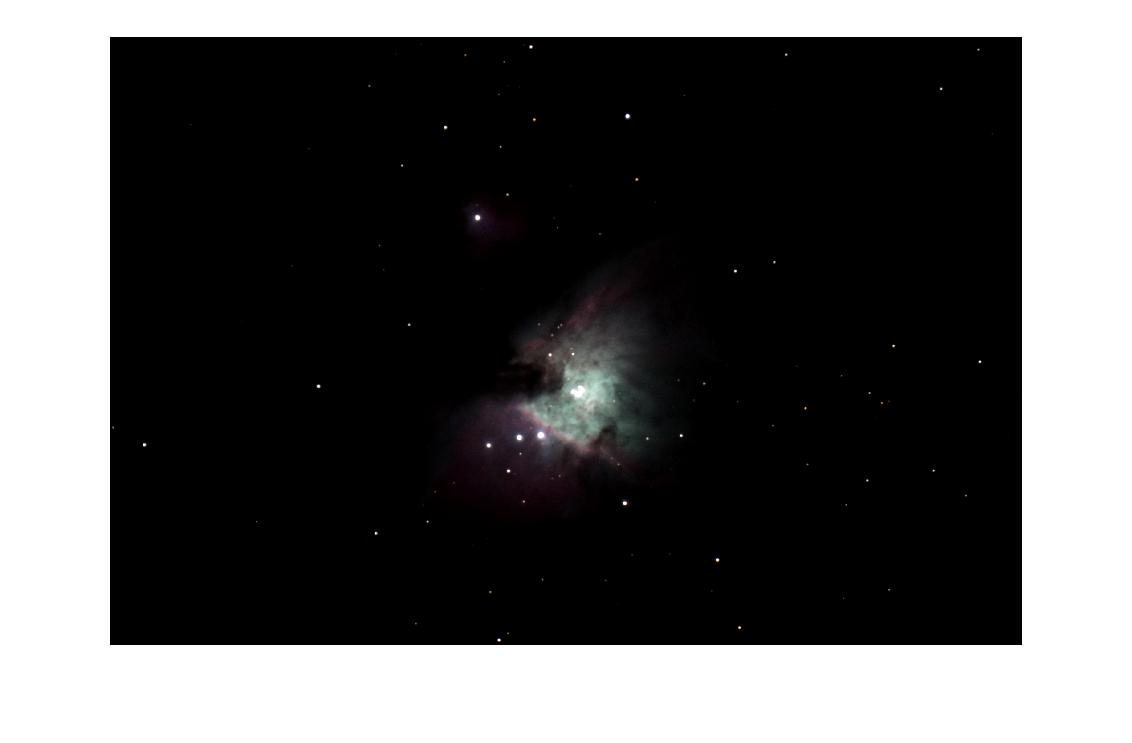
També he provat de fer la funció 1-x per tal que sortis la nebulosa i l’espai fos en blanc i el resultat és curiós:



A continuació, he provat la funció imadjust per a imatges amb color RGB de la següent manera(provant amb diferents paràmetres):

imadjust(Am,[0 0 0; 1 1 1],[])

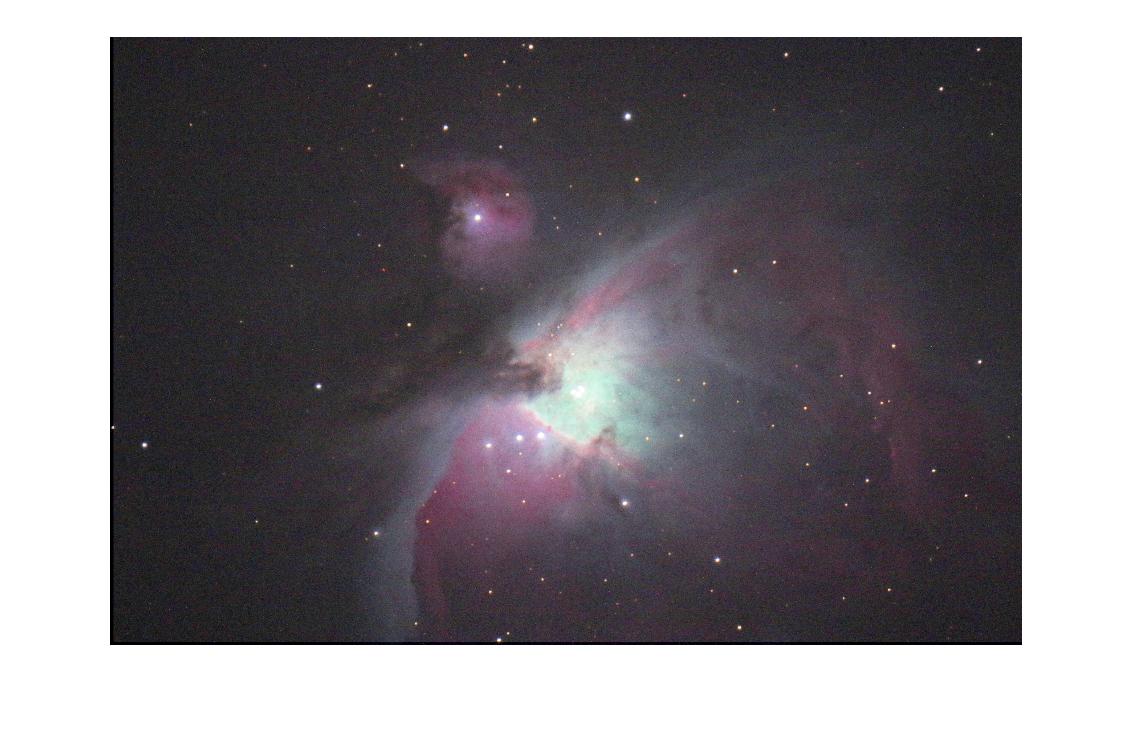
i el resultat queda força lluny del que volem:



Però la funcio que més s’acosta a l’objectiu és imlocalbrighten:

imlocalbrighten(Am);

Aquesta funció augmenta la il·luminació de les parts menys il·luminades de la imatge i, tot i que augmenta el soroll de la imate, el resultat és força satisfactori:



Finalment, després d’anar provant diferents paràmetres, diferents funcions i maneres de colocar aquestes funcions, el resultat que més m’ha agradat ha sigut utilitzant les dues funcions anteriors d’aquesta manera:

Am = imlocalbrighten(Am,0.7);

Am = imadjust(Am,[.1 .1 .1; 1 1 1],[]);

Així doncs, el resultat final(comparat amb l’inicial) que m’ha quedat ha sigut:

