

**FIB**Facultat d'Informàtica
de BarcelonaDepartament d'Enginyeria de Sistemes,
Automàtica i Informàtica Industrial

UNIVERSITAT POLITÈCNICA DE CATALUNYA

VISIÓ PER COMPUTADOR

Exercici 2 de Laboratori

Facultat d'Informàtica de Barcelona

Manel Frigola
Joan Climent

Barcelona, Febrer de 2021

Exercici 2.

Aquest exercici està basat en una simplificació de la contribució Astrophotography with MATLAB: Imaging the Orion Nebula per Loren Shure, 6 de Novembre 2020.

La idea de l'exercici es basa en fusionar dues imatges de la nebulosa d'Orion en una única imatge per aconseguir millor contrast i reducció del soroll (fig.1).

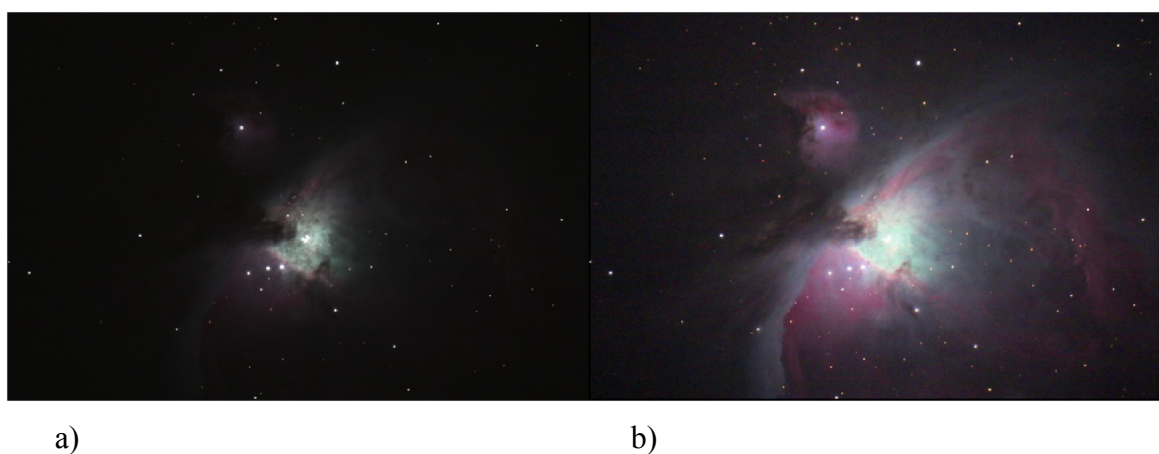


Fig. 1 Nebulosa d'Orion: a) Imatge original, b) Imatge resultant

Les imatges les trobareu annexes a aquest document.

1. El primer que farem és llegir les imatges i les convertim a *double*.

```
A = double(imread('_MG_7735.JPG'))/255;
```

```
B = double(imread('_MG_7737.JPG'))/255;
```

2. Comprovem que passaria si superposem les imatges directament. Per veure-ho, restem les imatges píxel a píxel i el resultat el re-escalem entre 0-1.

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

```
maxim = max(DIF(:));
```

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

```
imshow(DIF);
```

S'observa en la imatge diferència que les imatges apareixen mogudes una respecte a

l'altre. Això és degut a la rotació terrestre i a que les imatges han estat preses en instants de temps diferents. Entre una imatge i l'altre han transcorregut alguns minuts i en conseqüència les imatges apareixen desplaçades aproximadament 20 píxels en horitzontal i 20 píxels en vertical.

3. Traslladem la imatge B 20 píxels en diagonal i observem que les imatges s'ajusten prou bé.

```
Bd = imtranslate(B,[20, -20]);  
DIF = abs(A-Bd);  
maxim = max(DIF(:));  
DIF = DIF/maxim;  
imshow(DIF);
```

4. Ara ja podem sumar les dues imatge A i Bd per obtenir una nova imatge “millorada”.

```
Am = (A+Bd)/2; % imatge millorada  
Am = XXXXXX; % poseu aquí el que faci falta  
montage ({A,Am});
```

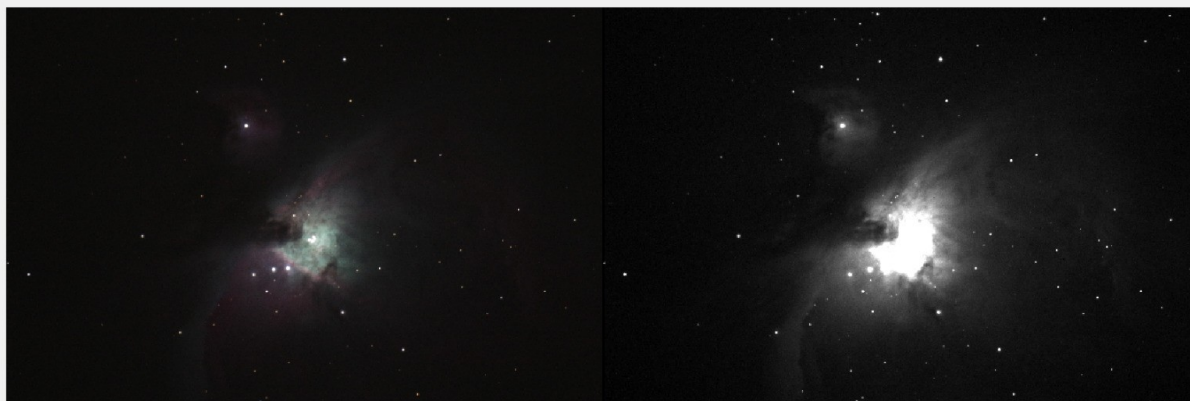
Proveu amb diferents funcions de contrast no lineal, que augmenti el contrast de les zones fosques de la imatge *Am* per posar de rellevància els colors de la nebulosa. En realitzar aquest contrast penseu que el color brillant de les estrelles no hauria de ‘pesar’ en el resultat. Feu un informe amb el que heu fet i entregueu-lo a Atenea.

Daniel Escribano

En totes les imatges següents, la imatge que es troba a l'esquerra representa la imatge original, sense aplicar cap modificació i la imatge de la dreta és la imatge després d'aplicar la funció que s'esmenta.

$Am = \text{imadjust}(\text{rgb2gray}(Am));$

Si fem servir la funció *imadjust* obtenim el següent resultat:



Com veiem a la imatge aquesta funció no és vàlida ja que realcem tant el contrast de las zones clares de la imatge com el de les zones fosques. Això passa ja que la funció satura el 1% dels valors dels píxels de menor intensitat i el 1% dels valors dels píxels amb major intensitat.

$Am = \text{histeq}(\text{rgb2gray}(Am));$

Si fem servir la funció *histeq* obtenim el següent resultat:

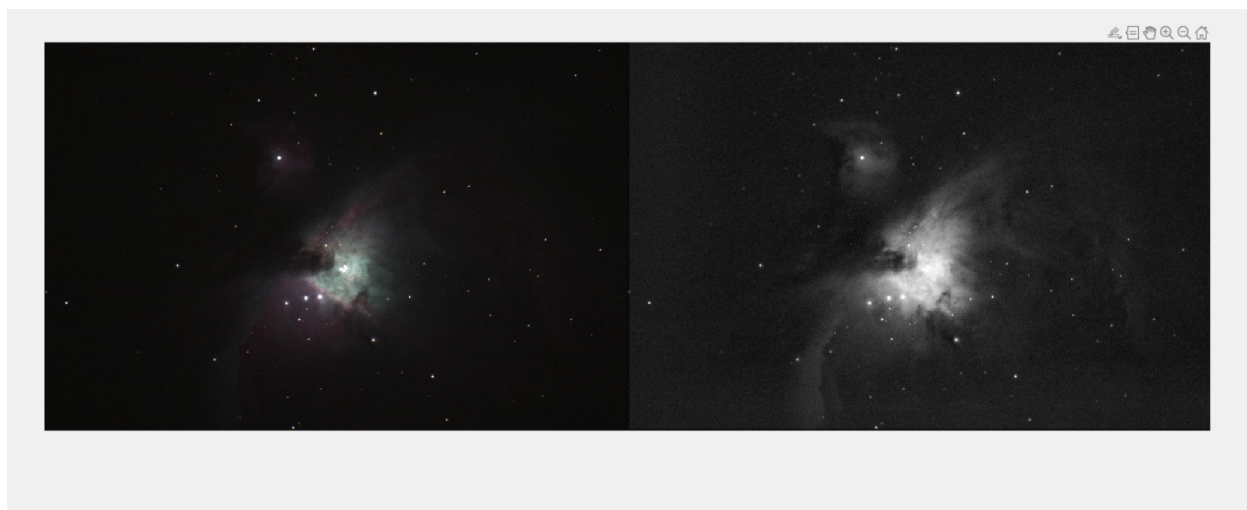


Com veiem a la imatge aquesta funció tampoc ens serveix. La funció *histeq* equalitza l'histograma corresponent als nivells de grisos de la imatge i ,quan no afegim cap paràmetre a la funció, modifica la imatge per tal de que el seu histograma representi 64 nivells de grisos distribuïts de forma uniforme.

Daniel Escribano

```
Am = adapthisteq(rgb2gray(Am));
```

Si fem servir la funció *adapthisteq* obtenim el següent resultat:



En fer servir aquesta funció obtenim un millor resultat que amb les dues funcions anteriors. Com veiem en la imatge, els colors de la nebulosa queden molt més realçats en comparació a la imatge original, tot i tenir una imatge en escala de grisos.

La funció *adapthisteq* aplica l'algorisme CLAHE(Contrast Limiting Adaptive Histogram Equalization) per realitzar una equalització adaptativa del histograma. L'algorisme CLAHE realitza la equalització en petites regions de la imatge i en acabar combina aquestes fent ús de interpolació bilinear (per tal de eliminar possibles vores a cada regió). Aquest algorisme limita la amplificació del contrast en cada regió, per tal de reduir el problema d'amplificació del soroll que hem pogut observar amb la funció *histeq*.

Les funcions anteriors totes treballen amb el supòsit de una imatge en *grayscale*, però si seguim el tutorial ofert per l'enllaç ofert podem obtenir una imatge en RGB on es trobin realçats els colors.

```
Am = rgb2hsv(Am);
```

```
L2 = log(Am(:,:,3));
```

```
blacklevel = -3.2;
```

```
L2 = rescale(L2, "InputMin", blacklevel);
```

```
Am(:,:,3) = L2;
```

```
img2 = hsv2rgb(Am);
```

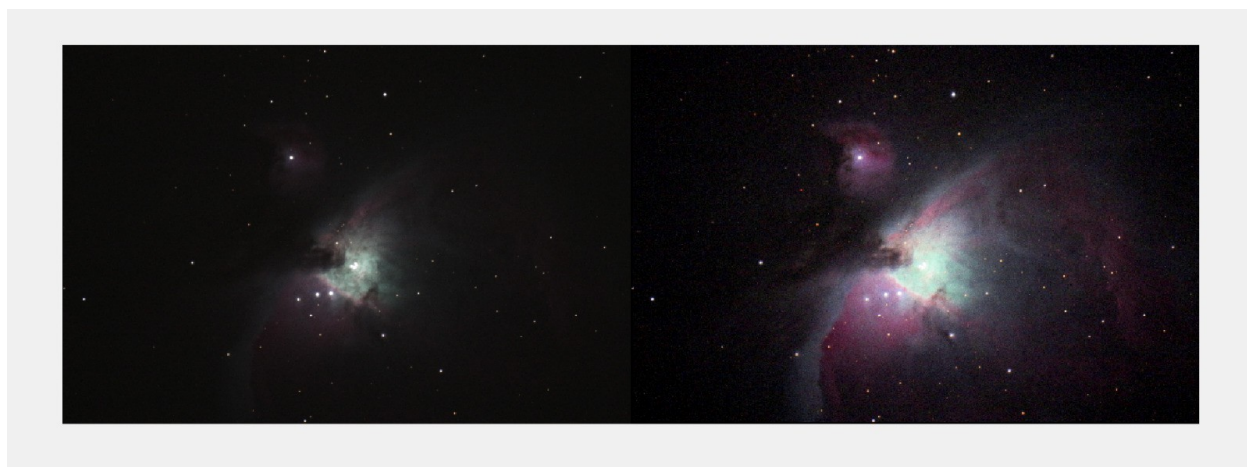


En aplicar aquest tros de codi podem obtenir la imatge superior. El gran inconvenient que ens trobem es que en l'enllaç usen una funció pròpia per passar de RGB al model HSL i MATLAB no ofereix una funció per realitzar aquesta conversió. Jo he fet servir el model HSV, però atès que aquest no es exactament igual que el model HSL (ja que el tercer component en HSV es troba en el rang del negre al color i en canvi el tercer component del model HSL va del negre al blanc) la imatge resultant no es exactament igual a la imatge b).

La ultima funció que he provat ha estat *imlocalbrighten* en combinació amb *imadjust*:

```
Am = imlocalbrighten(Am, 0.7);
```

```
Am = imadjust(Am, [.12 .12 .12; 1 1 1], []);
```



La funció *imlocalbrighten* ens proporciona una eina per il·luminar imatges amb poca lluminositat. Gràcies a la funció *imadjust* podem saturar els colors de la imatge resultant d'aplicar la funció anterior, especificant els l·lindars de contrast que volem aplicar a la imatge.

Daniel Escribano

Després de realitzar la pràctica, puc concloure que les formes de obtenir la imatge en RGB donen resultats satisfactoris. Tot i això, la funció *adapthisteq* també ens permet realçar els colors de la nebulosa respecte al fons negre de l'espai.