NemesisDP

Un programma automatico per la detection di supernovae extragalattiche

Marco Monaci - ISSP

La necessità di un software di detection

«il nostro budget anticollisione ci permette di controllare solo il 3% del cielo, e con tutto il rispetto signore il cosmo ha un culo enorme»

Dan Truman – Armageddon

Parafrasando l'affermazione di prima, possiamo dire che il cosmo è troppo grande per poterlo controllare tutto a mano e ad occhio.

Con un telescopio di medie dimensioni possiamo osservare oltre 50'000 galassie, e in ognuna delle quali esplode in media una supernova ogni secolo.

Sono davvero troppe. E' necessario automatizzare la ricerca.

Del resto, i computer sono stati creati apposta. Fare velocemente operazioni STUPIDE. In questo modo i nostri cervelli, molto più evoluti, possono pensare ad altro, invece di perdere tempo ad osservare centinaia di migliaia di galassie.

Questo non significa che non dobbiamo più controllarle. Se ci fa piacere, perché non farlo.

MA UNA COSA NON ESCLUDE L'ALTRA.

Fattibilità del progetto

Ovviamente, il progetto presenta difficoltà notevoli.

Sono necessarie molte funzioni, alcune delle quali piuttosto complesse, per ottenere una SICURA E CERTA detection.

Necessarie centinaia, se non migliaia di linee di codice.

Necessarie procedure di controllo, per evitare falsi allarmi.

Necessari sotto-controlli incrociati, per evitare falsi positivi come falsi negativi.

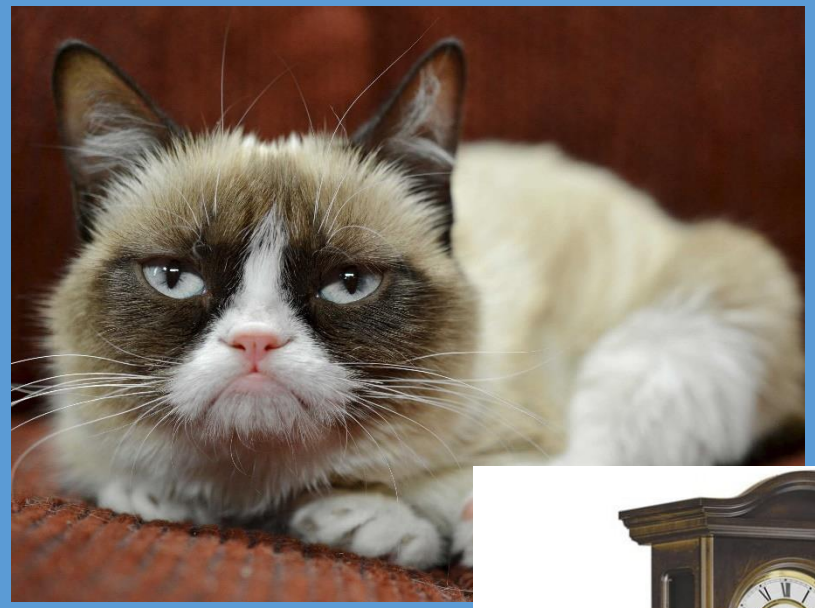
Sembra impossibile, vero?

ESATTO, COME IMPOSSIBILE SEMBRAVA NEGLI ANNI 40 PORTARE ESSERI UMANI SULLA LUNA.

Complesso o complicato?

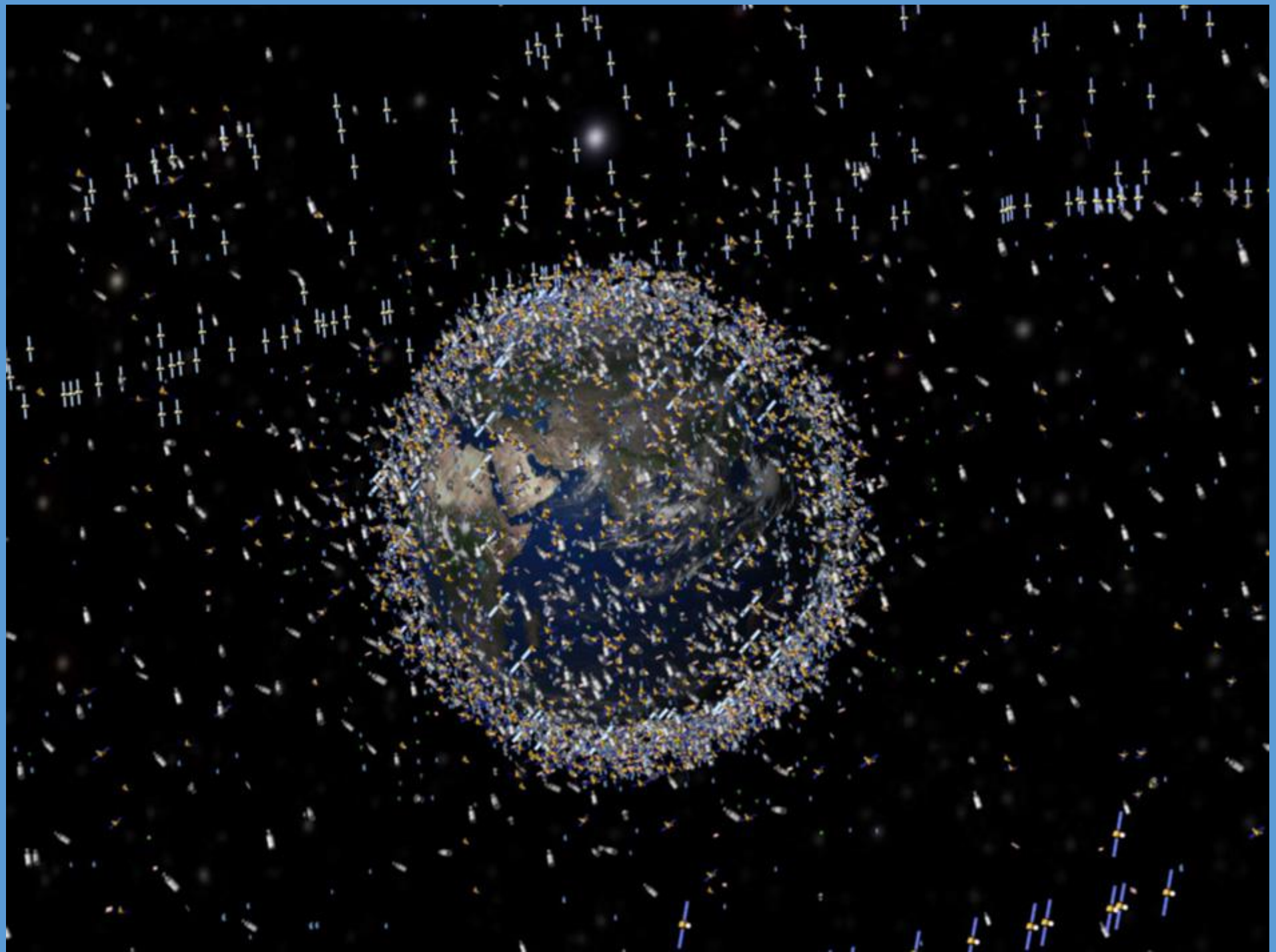
Immaginiamo di smontare un orologio in tutti i suoi componenti. Con fatica e lavoro, però, possiamo ricostruirlo. L'orologio è un oggetto COMPLICATO.

Immaginiamo ora di «smontare» un gatto in tutti i suoi componenti, fegato, stomaco, vasi sanguigni eccetera. Poi proviamo a «ricostruirlo». Per quanto ci sforziamo, il gatto non «funziona». Il gatto è un «oggetto» COMPLESSO.



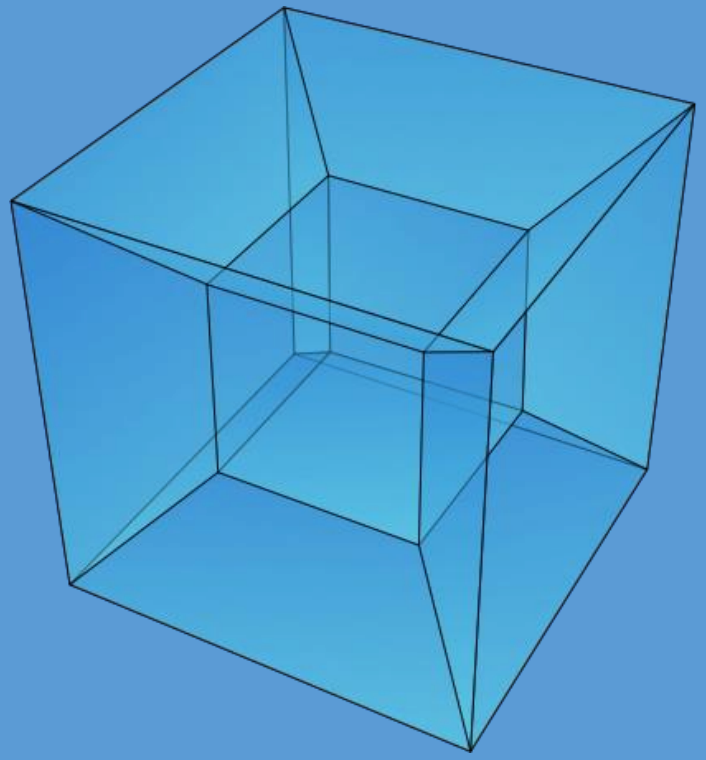
Ecco, il software possiamo ritenerlo ragionevolmente complicato. Per quanto sia intricata e nascosta, la soluzione c'è. A noi trovarla.

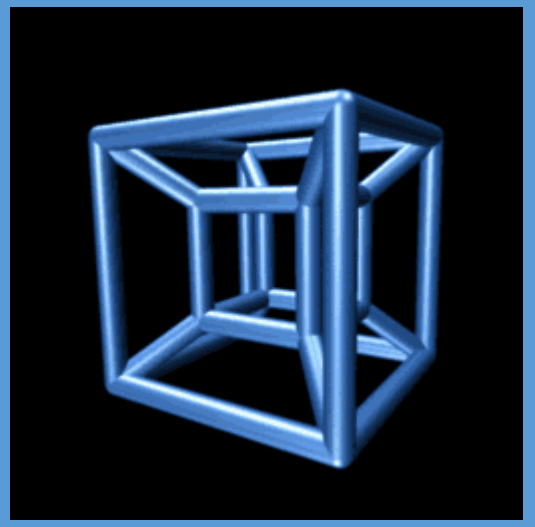
Del resto, ci sono software che eseguono lavori ben più complessi. Basti pensare ai software che dirigono il «traffico» satellitare.

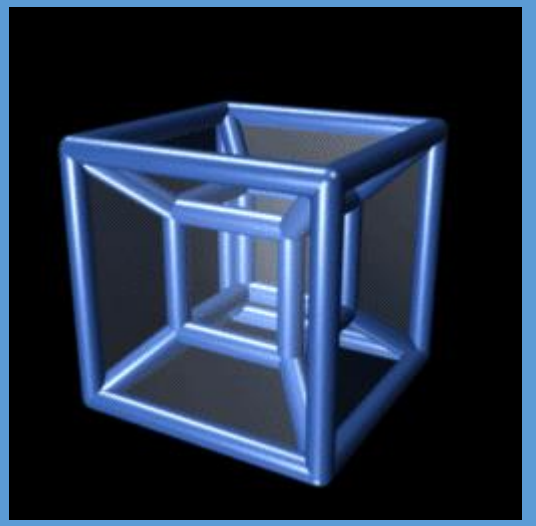


L’ambiente MatLab

MatLab sta per MatrixLaboratory. Il cuore di MatLab sono appunto le matrici, ovvero tabelle di numeri, anche multidimensionali. Tutto viene trasformato in matrici. Una matrice 1D sarà un vettore, ovvero una colonna di numeri, una matrice 2D sarà una tabella di numeri, una matrice 3D sarà un «cubo» pieno di numeri. Una matrice 4D è un ipercubo pieno di numeri, o se volete, un tesseratto (o un tensore n-dimensionale).







Come passare da una immagine ad una matrice

Non è difficile. L'immagine in bianco e nero è un enorme rettangolo formato da pixel. Quindi possiamo creare una matrice con tante righe quante sono le righe del CCD e tante colonne quante sono le colonne del CCD. Per ogni elemento mettiamo il valore in ADU del pixel stesso.

In questo modo abbandoniamo completamente l'idea di immagine e possiamo lavorare sulle matrici corrispondenti senza perdere alcuna informazione.

Alcune proprietà delle matrici

Le matrici hanno il notevole vantaggio di poter contenere grandi quantità di informazioni (dall'immagine alla matrice non perdiamo informazione) e di essere facili da gestire. Per esempio avrebbe poco senso moltiplicare una immagine per uno scalare (ovvero un valore costante), oppure avrebbe poco senso «sommare» due immagini.

Con le matrici tutto questo si può fare molto facilmente.

$$\begin{pmatrix}
0 & 3 & 1 & 0 & 2 & 3 & 8 & 1 & 1 & 3 \\
1 & 1 & 0 & 0 & 7 & 1 & 2 & 2 & 3 & 3 \\
1 & 2 & 2 & 0 & 0 & 6 & 7 & 1 & 2 & 2 \\
1 & 2 & 3 & 10 & 0 & 4 & 6 & 1 & 0 & 5 \\
3 & 2 & 2 & 1 & 4 & 3 & 2 & 1 & 6 & 0 \\
7 & 4 & 4 & 5 & 3 & 9 & 6 & 1 & 6 & 1 \\
7 & 1 & 1 & 5 & 2 & 8 & 9 & 1 & 3 & 6 \\
5 & 0 & 1 & 6 & 2 & 0 & 0 & 0 & 1 & 5 \\
1 & 6 & 3 & 3 & 4 & 6 & 2 & 0 & 1 & 1 \\
1 & 2 & 2 & 4 & 1 & 1 & 3 & 0 & 8 & 2
\end{pmatrix}$$

$$\begin{bmatrix} 2 & 3 & 5 & -4 \\ 4 & 8 & -2 & 5 \\ 3 & 7 & -1 & 8 \\ 1 & 1 & 6 & 2 \end{bmatrix} * 2 = \begin{bmatrix} 4 & 6 & 10 & -8 \\ 4 & 16 & -4 & 10 \\ 6 & 14 & -2 & 16 \\ 2 & 2 & 12 & 4 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 3 & 5 & -4 \\ 4 & 8 & -2 & 5 \\ 3 & 7 & -1 & 8 \\ 1 & 1 & 6 & 2 \end{bmatrix} + \begin{bmatrix} 2 & 5 & 7 & -1 \\ -3 & 5 & -3 & 2 \\ 2 & 9 & -1 & 9 \\ 4 & -1 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 4 & 7 & 12 & -5 \\ 1 & 13 & -5 & 7 \\ 5 & 16 & -2 & 17 \\ 5 & 0 & 6 & 4 \end{bmatrix}$$

$$
\begin{aligned}
\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}
\begin{pmatrix} a & d \\ b & e \\ c & f \end{pmatrix}
&= \begin{pmatrix} 1 \\ 4 \\ 7 \end{pmatrix} \begin{pmatrix} a & d \end{pmatrix} + \begin{pmatrix} 2 \\ 5 \\ 8 \end{pmatrix} \begin{pmatrix} b & e \end{pmatrix} + \begin{pmatrix} 3 \\ 6 \\ 9 \end{pmatrix} \begin{pmatrix} c & f \end{pmatrix} \\
&= \begin{pmatrix} 1a & 1d \\ 4a & 4d \\ 7a & 7d \end{pmatrix} + \begin{pmatrix} 2b & 2e \\ 5b & 5e \\ 8b & 8e \end{pmatrix} + \begin{pmatrix} 3c & 3f \\ 6c & 6f \\ 9c & 9f \end{pmatrix} \\
&= \begin{pmatrix} 1a+2b+3c & 1d+2e+3f \\ 4a+5b+6c & 4d+5e+6f \\ 7a+8b+9c & 7d+8e+9f \end{pmatrix}.
\end{aligned}
$$

Le idee alla base del software

Per ora le (poche) idee possono essere riassunte in questo prospetto:

- Caricamento delle immagini manualmente (manca funzione)
- Calcolo della FFT (Fast Fourier Transorm)
- Applicazione di un filtro passa basso per eliminare il rumore termico
- Applicazione della funzione clean.m
- Sottrazione delle immagini
- Individuazione dei massimi

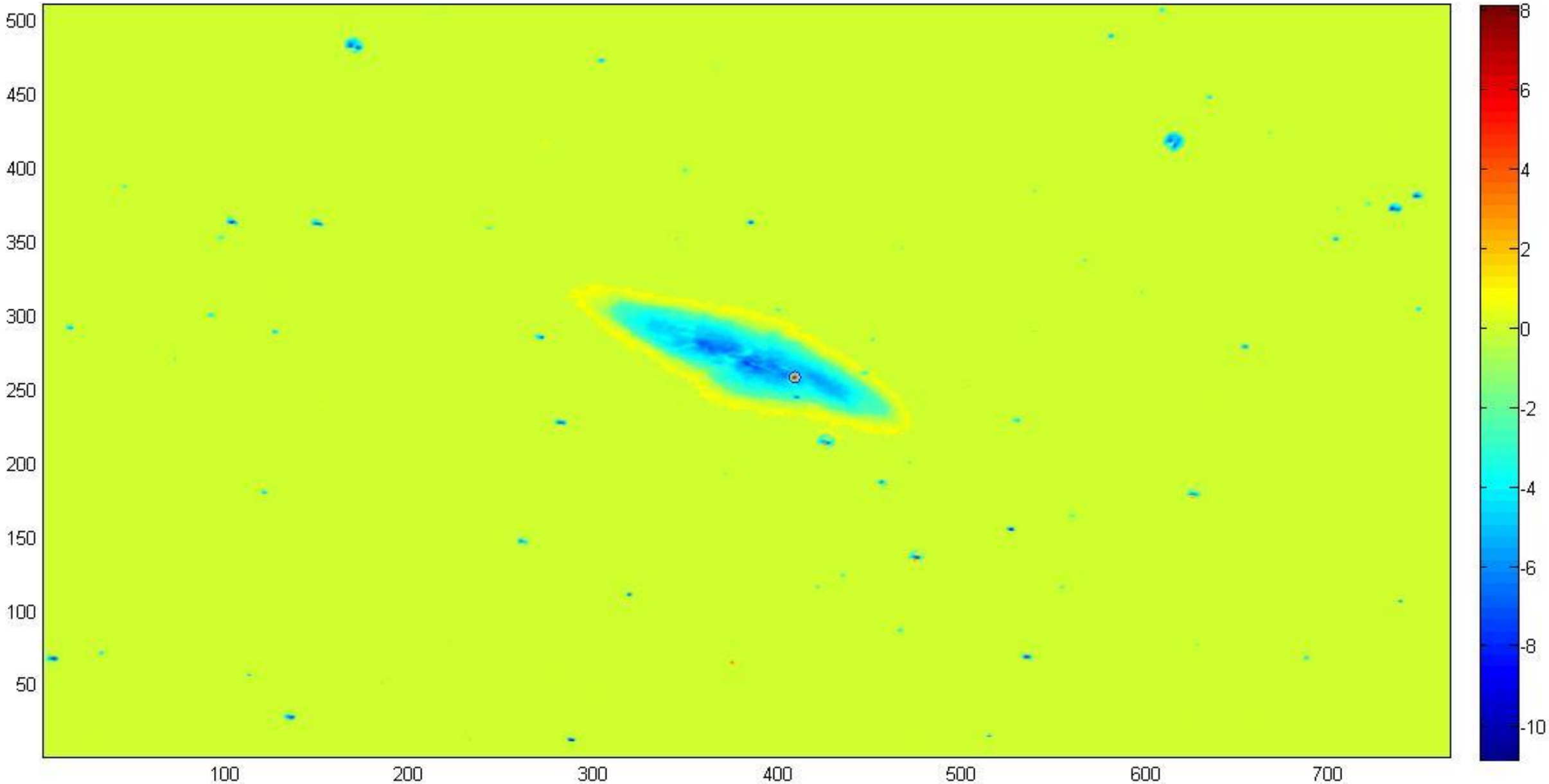
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%                   Nemesis DP                      %
% Programma per il riconoscimento automatico di SN  %
%           Scritto da Marco Monaci                 %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%pulizia dello schermo e variabili
%---------------------------------
clear all;
close all;
clc;

%caricamento immagini da analizzare e di riferimento
%---------------------------------------------------
imm=fitsread('NGC3034_SN.fit');
ref=fitsread('NGC3034.fit');
```

```
20      %-------------------------------------------------------------
21 -    fourier1=fft2(imm);
22 -    fourier2=fft2(ref);
23 -    [M,N]=size(imm);
24 -    [P,Q]=size(ref);
25 -    u=0:(M-1);
26 -    v=0:(N-1);
27 -    x=0:(P-1);
28 -    y=0:(Q-1);
29 -    idx = find(u>M/2);
30 -    u(idx) = u(idx)-M;
31 -    idy = find(v>N/2);
32 -    v(idy) = v(idy)-N;
33 -    [V,U] = meshgrid(v,u);
34 -    edx=find(x>P/2);
35 -    x(edx)=x(edx)-P;
36 -    edy=find(y>Q/2);
37 -    y(edy)=y(edy)-Q;
38 -    [Y,X]=meshgrid(y,x);
39 -    d0 = 150;          % Frequenza di taglio per l'immagine
40 -    D0 = sqrt(U.^2+V.^2);
41 -    H_LP_GAUSS1 = exp(-(D0.^2)./(2*(d0^2)));
42 -    G1 = H_LP_GAUSS1.*fourier1;
43 -    passaBassoImm = real(ifft2(G1));
44 -    d1 = 160;          % Frequenza di taglio per il riferimento
45 -    D1 = sqrt(Y.^2+X.^2);
46 -    H_LP_GAUSS2 = exp(-(D1.^2)./(2*(d1^2)));
47 -    G2 = H_LP_GAUSS2.*fourier2;
48 -    passaBassoRef = real(ifft2(G2));
```

```
50      %rescale
51      %-------------------------------------------------------
52 -    [imm,ref]=rescale((passaBassoImm),(passaBassoRef));
53
54      %pulizia immagini tramite funzione clean.m
55      %-------------------------------------------------------
56 -    [ImmCleaned]=clean(imm, 1);
57 -    [RefCleaned]=clean(ref, 1);
58
59 -    pcolor(ImmCleaned), shading interp;
60 -    figure;
61 -    pcolor(RefCleaned), shading interp;
62 -    figure;
63
64      %funzione allineamento immagini
65      %ImmAligned=ImmCleaned;
66      %optimizer = registration.optimizer.OnePlusOneEvolutionary;
67      %metric = registration.metric.MattesMutualInformation;
68      %RefAligned=imregister(passaBassoRef, ImmAligned, 'affine', optimizer, metric);
69
70 -    sub=ImmCleaned-RefCleaned;
71 -    subpos=max(0,sub);
```

```
64      %funzione allineamento immagini
65      %ImmAligned=ImmCleaned;
66      %optimizer = registration.optimizer.OnePlusOneEvolutionary;
67      %metric = registration.metric.MattesMutualInformation;
68      %RefAligned=imregister(passaBassoRef, ImmAligned, 'affine', optimizer, metric);
69
70 -    sub=ImmCleaned-RefCleaned;
71 -    subpos=max(0,sub);
72
73
74      %individuazione del massimo della matrice subpos e posizionamento marker
75      %-------------------------------------------------------------------
76 -    maximum=max(max(subpos));
77 -    [i,j]=find(subpos==maximum);
78 -    immlog=20*log(imm);
79 -    pcolor(immlog),shading interp;
80 -    hold on;
81 -    plot(j,i,'ro','MarkerSize',15,'LineWidth',3);
82
```

```
%la funzione pulisce l'immagine sottraendo la media più n volte la
%deviazione standard della matrice e tutto ciò che risulta negativo viene posto uguale a 1
function [final]=clean(A, n)
r=size(A,1);
c=size(A,2);
media=mean(mean(A));
dev=mean(std(A));
elab=ones(r,c)*(media+(n*dev));
sott=A-elab;
for i=1:c
    for t=1:r
        if sott(t,i)<0
            sott(t,i)=1;
        end
    end
end
final=sott;
```


500
450
400
350
300
250
200
150
100
50
100
200
300
400
500
600
700
8
6
4
2
0
-2
-4
-6
-8
-10

Qualche prova con Matlab