

Calcolo scientifico

Marco Miani

22 agosto 2018

Indice

1 Scegliere la porzione di immagine	2
2 Sfocare l'immagine	3
3 Rimettere a fuoco l'immagine	4
3.1 Condizioni Periodiche	5
3.2 Condizioni di Neumann	5
3.3 Regolarizzazione	5
3.4 Risultati grafici	7
3.4.1 PSF esponenziale	8
3.4.2 PSF out-of-focus	9
4 Plot rapporto autovalori	10

Introduzione

Per la sperimentazione ho scattato una foto in giardino, la allego come file 'cactus.jpg', di dimensioni 4032×3024 , e la riporto a fondo pagina. Seguendo le orme dell'articolo l'ho sfocata tramite convoluzione utilizzando due tipi di PSF: esponenziale e out-of-focus. *Successivamente*, da ciascuna delle due foto ottenute ho ritagliato una porzione, della stessa zona per poterle poi confrontare, di dimensioni 256×256 in modo da rendere più agevole la computazione (avendo scelto convoluzione di tipo 'valid' sarebbe stato possibile, in linea teorica, operare anche sull'intera immagine sfocata). Dopodiché ho rimesso a fuoco entrambe le immagini. Per sopperire alla mancanza di informazioni al bordo, ho imposto prima condizioni periodiche e dopo condizioni di Neumann.

Ho inoltre aggiunto un parametro di regolarizzazione per limitare l'amplificazione del rumore. Ho fatto delle prove per verificare l'ottimalità del parametro fornito dall'articolo e, in caso di risposta negativa, per ottenere il vero parametro ottimale.

Alla fine, come richiesto, ho fatto il plot del rapporto tra gli autovalori di C e N , le matrici ottenute con condizioni periodiche e di Neumann rispettivamente, per vedere quanto differiscono spettralmente le matrici dei due modelli.



1 Scegliere la porzione di immagine

Questa è stata sicuramente la parte più facile, ma non per questo la meno importante. Ho scelto il quadrato 256×256 (in figura 1) con angolo in alto a sinistra nel punto (1650, 3150) perché questa porzione di immagine offriva vari dettagli di diverso tipo, molto utili nel verificare la bontà dei risultati finali. Inoltre presenta una molto vaga periodicità lungo l'asse y, e una totale assenza di periodicità lungo l'asse x che portano poi, nei test con condizioni periodiche, ad una cornice leggermente più rovinata nei lati verticali che nei lati orizzontali.



Figura 1: Immagine messa a fuoco iniziale

2 Sfocare l'immagine

Ho generato le due Point Spread Function di centro $h_{0,0}$ come matrici 17×17 . Come suggeriva l'articolo, per la PSF esponenziale ho usato:

$$h_{i,j} = \begin{cases} c_{exp} e^{-0.1((i-8)^2 + (j-8)^2)} & \text{se } |i-j| \leq 8 \\ 0 & \text{altrimenti} \end{cases}$$

mentre per la PSF out-of-focus:

$$h_{i,j} = \begin{cases} c_{oof} & \text{se } (i-8)^2 + (j-8)^2 \leq 4^2 \\ 0 & \text{altrimenti} \end{cases}$$

dove c_{exp} e c_{oof} sono tali che la somma faccia 1. Allego le function Matlab 'psf_exp.m' e 'psf_oof.m' che generano tali matrici. Qui sotto riporto i risultati

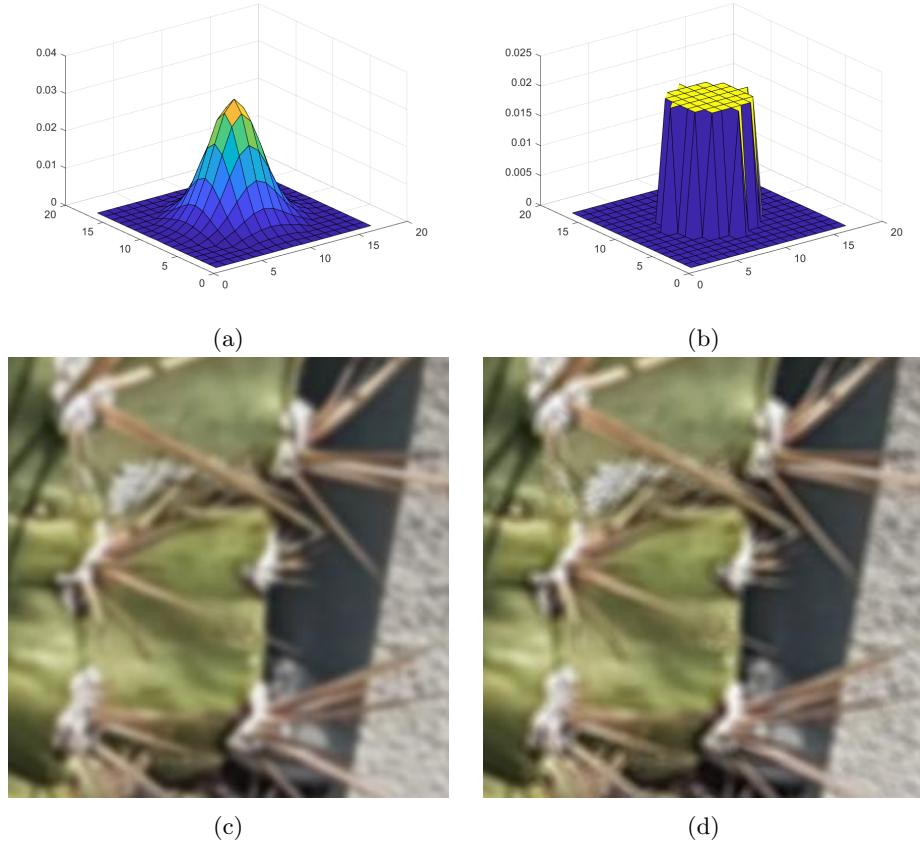


Figura 2: Sfocatura Esponenziale (a) e relativa immagine ottenuta (c)
Sfocatura out-of-focus (b) e relativa immagine ottenuta (d)

Il codice Matlab usato per sfocare l'immagine tramite convoluzione

```
function b = sfoca(a, psf)
% sfoca un'immagine memorizzata nella variabile a
% usando la PSF memorizzata nella variabile psf
% in uscita l'immagine sfocata b
a = double(a);
s = size(a);
if length(s)==3
    b1 = conv2(a(:,:,1), psf, 'valid');
    b2 = conv2(a(:,:,2), psf, 'valid');
    b3 = conv2(a(:,:,3), psf, 'valid');
    b = b1;
    b(:,:,2) = b2;
    b(:,:,3) = b3;
else
    b = conv2(a, psf, 'valid');
end
end
```

3 Rimettere a fuoco l'immagine

Un'immagine sfocata "prende informazioni" da un area più ampia di tanto quanto è il raggio della PSF, in ogni direzione. Per sopperire alla mancanza di informazione dobbiamo fare delle assunzioni sull'intorno dell'immagine, chiamate condizioni al contorno. Riducendo il problema a risolvere il sistema $Ax = b$.

Abbiamo assunto prima condizioni periodiche, giustapponendo l'immagine a se stessa, e poi condizioni di Neumann, giustapponendo l'immagine *ribaltata* a se stessa, così da avere una sorta di derivata = 0 lungo la perpendicolare ai bordi.

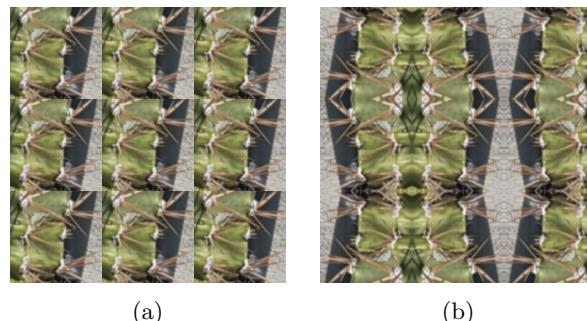


Figura 3: Esempio intuitivo di condizioni di condizioni Periodiche (a) e di Neumann (b). Ovviamente nella pratica manipoleremo le matrici di sfocatura

3.1 Condizioni Periodiche

Imponendo condizioni periodiche otteniamo una matrice di sfocatura circolante, queste matrici hanno la caratteristica di essere diagonalizzabili tramite la trasformata discreta di Fourier (DFT) e la sua inversa. Questo ci permette di risolvere il sistema in $O(n \log n)$, dove n è il numero di pixel dell'immagine. Nel caso unidimensionale con PSF = $[h_m, \dots, h_0, \dots, h_{-m}]$ la matrice è

$$A = \begin{pmatrix} 0 & h_{-m} & \cdots & h_1 \\ \ddots & \ddots & & \vdots \\ h_m & 0 & h_{-m} & \\ \vdots & \ddots & \ddots & \\ h_1 & \cdots & h_m & 0 \end{pmatrix} + \begin{pmatrix} h_0 & \cdots & h_{-m} & 0 \\ \vdots & \ddots & & \ddots \\ h_m & & \ddots & h_{-m} \\ 0 & \ddots & h_m & \cdots & h_0 \end{pmatrix}$$

dove la parte sinistra è il contributo "dell'esterno dell'immagine". Nel caso bidimensionale, indicando l'immagine sfocata con il vettore b ottenuto giustappponendo le colonne di pixel ($b = immagine(:)$ in Matlab), otteniamo una matrice circolante a blocchi con blocchi circolanti che si diagonalizza tramite trasformata di Fourier bidimensionale.

3.2 Condizioni di Neumann

Imponendo condizioni di Neumann otteniamo una matrice di sfocatura di Toeplitz-plus-Hankel, queste matrici hanno la caratteristica di essere diagonalizzabili tramite la trasformata discreta del coseno (DDT) e la sua inversa. Anche qui, quindi, possiamo risolvere il sistema in $O(n \log n)$. Nel caso unidimensionale è

$$A = \begin{pmatrix} h_1 & \cdots & h_m & 0 \\ \vdots & \ddots & & \ddots \\ h_m & 0 & h_{-m} & \\ \ddots & \ddots & \ddots & \vdots \\ 0 & h_{-m} & \cdots & h_1 \end{pmatrix} + \begin{pmatrix} h_0 & \cdots & h_{-m} & 0 \\ \vdots & \ddots & & \ddots \\ h_m & & \ddots & h_{-m} \\ 0 & \ddots & h_m & \cdots & h_0 \end{pmatrix}$$

dove, come prima, la parte sinistra è il contributo dato dalle condizioni al contorno. Nel caso bidimensionale otteniamo una matrice di Toeplitz-plus-Hankel a blocchi con blocchi di Toeplitz-plus-Hankel, diagonalizzabile tramite trasformata del coseno bidimensionale.

3.3 Regolarizzazione

Invece di risolvere il sistema $Ax = b$ risolviamo il sistema

$$(AA^T + \alpha I)w = b$$

e poi otteniamo $x = A^Tw$. In questo modo, facendo variare il parametro α , allontaniamo gli autovalori da 0 e riduciamo così la propagazione dell'errore.

Della matrice A in ogni caso siamo interessati solo alla prima colonna (il cui reshape a matrice quadrata chiamiamo ' a ') che determina univocamente tutta la matrice A e ci basta per ottenerne gli autovalori. Nel caso periodico abbiamo

$$a = \begin{pmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,m} & 0 & h_{0,-m} & \cdots & h_{0,-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,m} & & h_{1,-m} & \cdots & h_{1,-1} \\ \vdots & \vdots & \ddots & \vdots & & \vdots & & \vdots \\ h_{m,0} & h_{m,1} & \cdots & h_{m,m} & & h_{m,-m} & \cdots & h_{m,-1} \\ 0 & & & & 0 & & & 0 \\ h_{-m,0} & h_{-m,1} & \cdots & h_{-m,m} & & h_{-m,-m} & \cdots & h_{-m,-1} \\ \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ h_{-1,0} & h_{-1,1} & \cdots & h_{-1,m} & 0 & h_{-1,-m} & \cdots & h_{-1,-1} \end{pmatrix}$$

Nel caso di Neumann invece abbiamo

$$a = \begin{pmatrix} h_{0,0} + h_{0,-1} + h_{-1,0} + h_{-1,-1} & h_{0,1} + h_{0,-2} + h_{-1,1} + h_{-1,-2} & \cdots & h_{0,m} + h_{-1,m} & 0 \\ h_{1,0} + h_{1,-1} + h_{-2,0} + h_{-2,-1} & h_{1,1} + h_{1,-2} + h_{-2,1} + h_{-2,-2} & \cdots & h_{1,m} + h_{-2,m} & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ h_{m,0} + h_{m,-1} & h_{m,1} + h_{m,-2} & \cdots & h_{m,m} & 0 \\ 0 & & \cdots & & 0 \end{pmatrix}$$

con tanti 0 da far combaciare la dimensione con quella di b .

Qui sotto la function Matlab che fa quanto richiesto, genera la matrice a e poi risolve il sistema.

```
function x = invert(b, psf, type, reg)
% risolve il sistema Ax=b
% dove A e la matrice di sfocatura ottenuta dalla psf
% type puo essere 'periodic' o 'neumann'
% reg e il parametro di regolarizzazione
[m,n] = size(psf); %la psf e sempre quadrata
meta = floor(m/2);
[mb,nb] = size(b);
if strcmp(type, 'periodic')
    %genero la matrice a
    a = [ psf zeros(m,nb-n) ; zeros(mb-m,nb) ];
    traslm = [meta+1:mb 1:meta];
    trasln = [meta+1:nb 1:meta];
    a = a(traslm,trasln);
    %calcolo la soluzione del sistema
    aat = ifft2( fft2(a) .* fft2(a) );
    aat(1,1) = aat(1,1) + reg;
    w = ifft2( fft2(b) ./ fft2(aat) );
    x = ifft2( fft2(w) .* fft2(a) );
```

```

end
if strcmp(type, 'neumann')
    %isolo , ruoto e rifletto le componenti della psf
    psfNW = rot90(psf(1:meta, 1:meta),2);
    psfNE = flipud(psf(1:meta, meta+1:end));
    psfSW = fliplr(psf(meta+1:end, 1:meta));
    psfSE = psf(meta+1:end, meta+1:end);
    %aggiusto le dimensioni
    psfNW(meta+1,meta+1) = 0;
    psfNE(meta+1,meta+1) = 0;
    psfSW(meta+1,meta+1) = 0;
    %genero la matrice a
    a = psfNW + psfNE + psfSW + psfSE;
    a = [ a zeros(meta+1,nb-meta-1) ; zeros(mb-meta-1,nb) ];
    e1 = zeros(mb,nb);
    e1(1,1)=1;
    %calcolo la soluzione del sistema
    dct2e1 = dct2(e1);
    aat = idct2( dct2(a) .* (dct2(a)./dct2e1) );
    aat(1,1) = aat(1,1) + reg;
    w = idct2( dct2(b) ./ (dct2(aat)./dct2e1) );
    x = idct2( dct2(w) .* (dct2(a)./dct2e1) );
end
end

```

Di fatto sommare αI , lavorando solo con la prima colonna di A , si riduce a sommare α ad $a(1,1)$.

In effetti la vera function utilizzata (la allego come 'invert.m') distingue i casi $reg = 0$ e $reg \neq 0$ per evitare la perdita non necessaria di informazioni, nel primo caso infatti risolve direttamente il sistema $Ax = b$ con

$$x = ifft2(fft2(b)./fft2(a)); \text{ nel caso periodico}$$

$$x = idct2\left(dct2(b)./\frac{(dct2(a))}{dct2(e1)}\right); \text{ nel caso di Neumann}$$

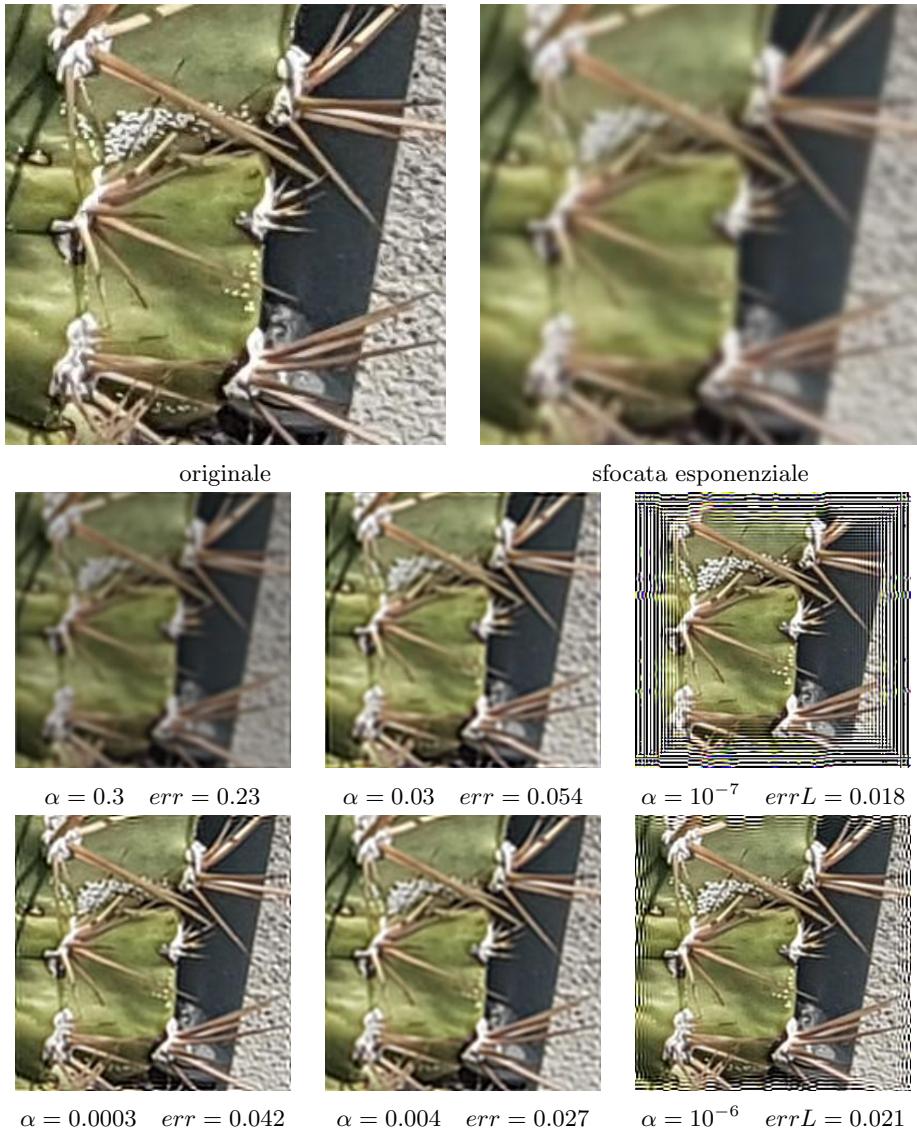
3.4 Risultati grafici

Nelle prossime pagine riporto i risultati confrontati. Per entrambi i tipi di PSF le tabelle contengono: porzione di immagine originale e sfocata (in alto), i risultati imponendo condizioni periodiche (al centro), i risultati imponendo condizioni di Neumann (in basso). Per stimare la bontà della soluzione si usa l'errore relativo

$$err = \frac{\| originale - rifocata \|_2}{\| originale \|_2}$$

Nei risultati a sinistra ho usato l' α suggerito nell'articolo, in quelli al centro l' α che minimizza l'errore relativo err , in quelli a destra l' α che minimizza l'errore relativo limitato al centro dell'immagine $errL$, cioè l'errore calcolato sull'immagine a cui è stata rimossa una cornice di 50 pixel.

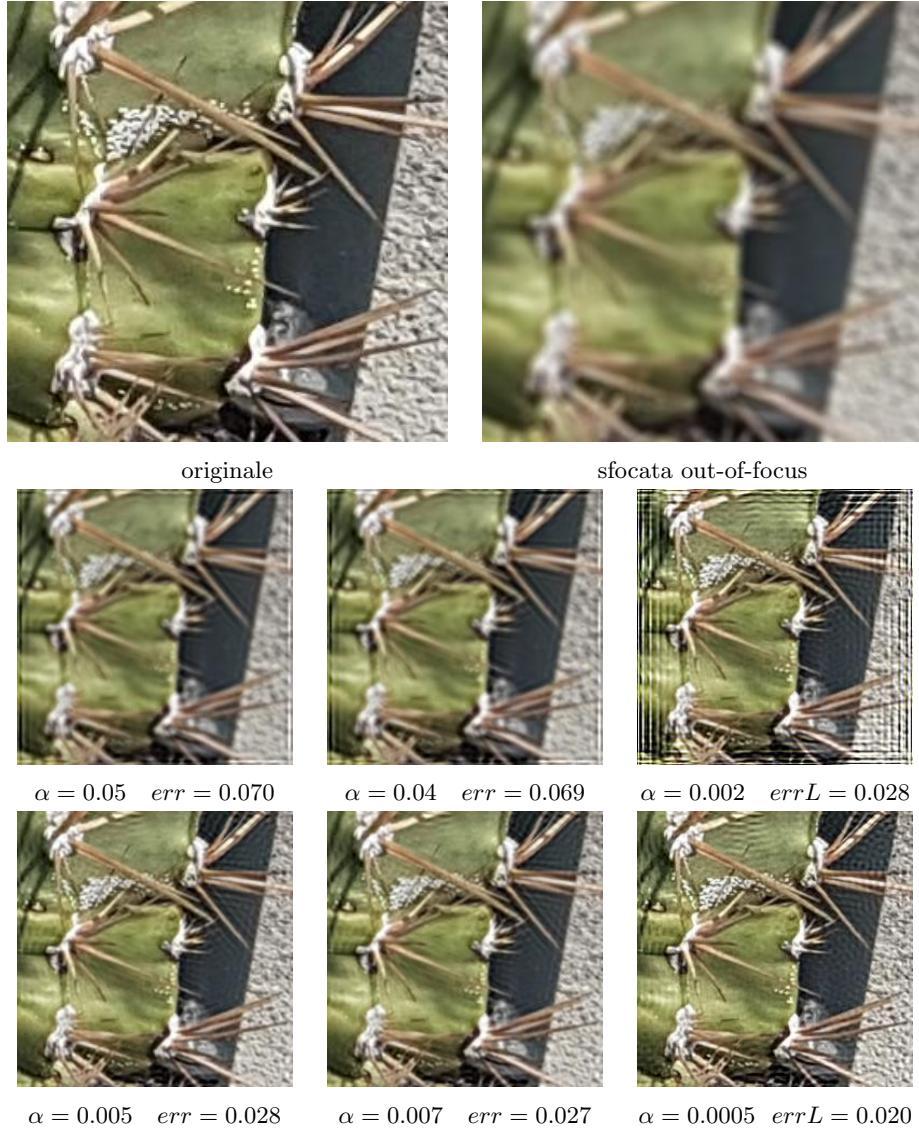
3.4.1 PSF esponenziale



L' α dell'articolo per condizioni periodiche (0.3) non è affatto ottimale, infatti si può migliorare di quasi un ordine di grandezza, forse è stata semplicemente una svista tipografica.

L' α suggerito per le condizioni di Neumann (0.0003) invece sembra essere una mediazione tra l'ottimizzare la figura e non rovinare troppo la cornice, il risultato in effetti è un buon compromesso.

3.4.2 PSF out-of-focus



In questo caso gli α suggeriti dall'articolo sono entrambi molto buoni, anche se si possono ottenere immagini decisamente più definite a discapito della qualità della cornice.

In questo caso si può apprezzare a pieno l'inefficienza delle condizioni periodiche (al centro) contro le condizioni di Neumann (in basso), queste ultime infatti limitano notevolmente il rumore al contorno, nonostante i parametri di regolarizzazione usati per queste siano minori.

Lo script in allegato 'test.m' provvede poi ad utilizzare le funzioni in modo appropriato, a leggere l'immagine da file, a salvare le immagini ottenute e a plottare i grafici prodotti. Inoltre misura il tempo impiegato nei vari passaggi.

4 Plot rapporto autovalori

In modo analogo alla funzione 'invert.m' si costruiscono le matrici di sfocatura C (cyclante) e N (di Neumann) e se ne calcolano gli autovalori tramite trasformata discreta e del coseno rispettivamente. Successivamente se ne fa il rapporto e se ne prende la norma elemento per elemento.

Questo procedimento è svolto dalla funzione 'fun.m' che riporto nelle sue sole parti interessanti.

```
function x = fun(mb, nb, psf)
% x = norma autovalori condizioni periodiche/neumann
% prende in input la psf e le dimensioni [mb,nb] della
% matrice a cui si devono poter applicare le matrici
% di sfocatura generate

% ... genera la matrice 'a' per le condizioni periodiche
autovaloriC = fft2(a);
% ... genera la matrice 'a' per le condizioni di Neumann
autovaloriN = (dct2(a)./dct2(e1));

xx = autovaloriC./autovaloriN;
x = sqrt(xx.*conj(xx));
end
```

Qui sotto riporto invece la parte dello script 'test.m' che completa i dettagli sul plot degli autovalori

```
rapporto_autovalori_exp = fun(m,n,psf_e);
figure; mesh(rapporto_autovalori_exp);
set(gca, 'ZScale', 'log');
title('AutovaloriC/AutovaloriN--PSF-esponenziale');

rapporto_autovalori_oof = fun(m,n,psf_o);
figure; mesh(rapporto_autovalori_oof);
set(gca, 'ZScale', 'log');
title('AutovaloriC/AutovaloriN--PSF-out-of-focus');
```

Nella prossima pagina, in figura 6, i plot così ottenuti

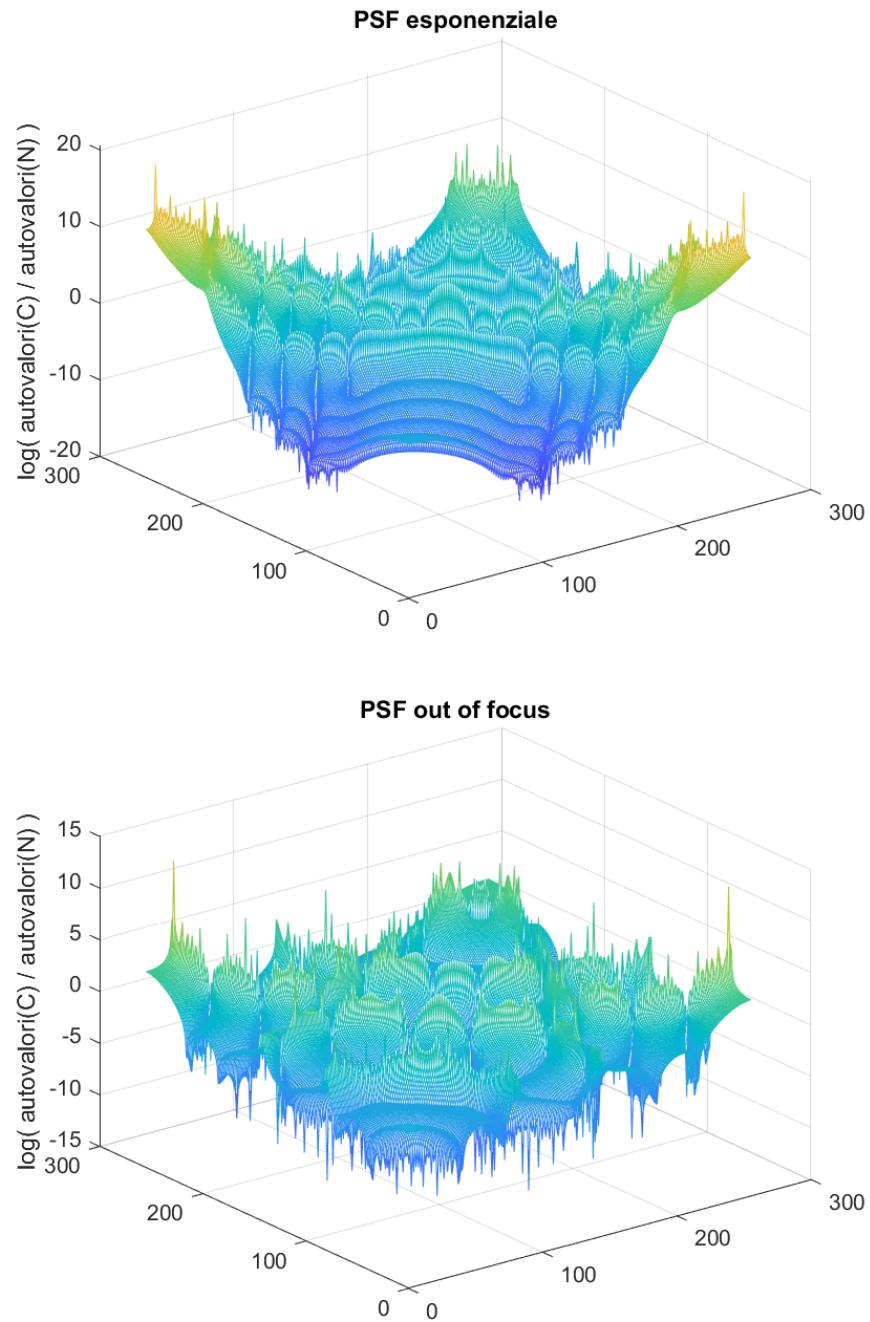


Figura 6: Rapporto tra gli autovalori delle matrici di sfocatura, in scala logaritmica. I plot così ottenuti però sono ben poco leggibili.

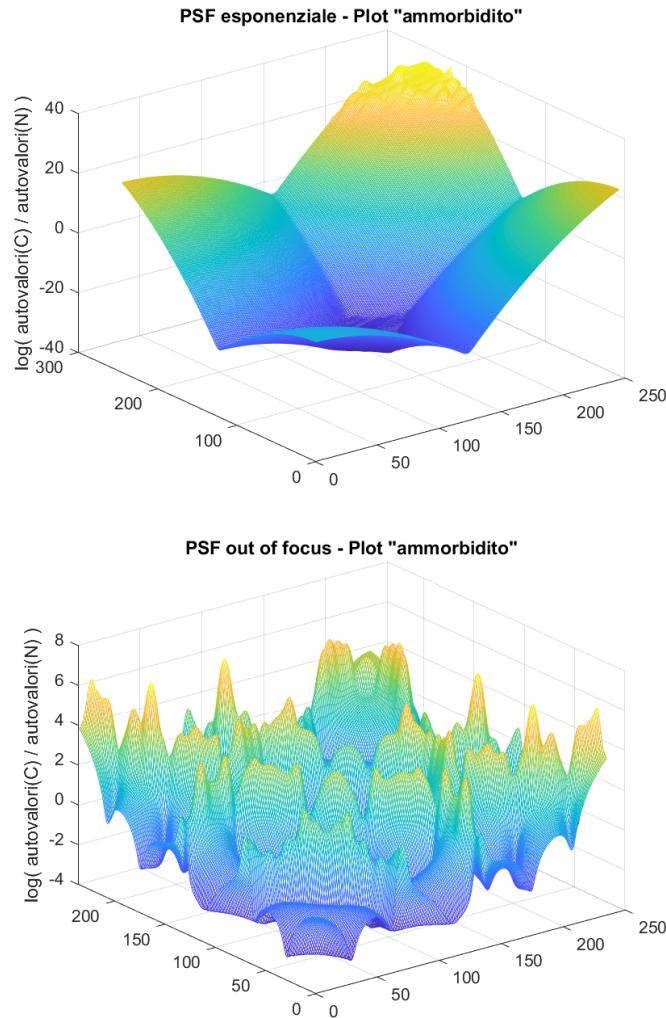


Figura 7: Per rimediare alla bassa leggibilità dei grafici precedenti ho pensato di eliminare la restrizione al rombo del supporto della PSF esponenziale, e poi di riutilizzare la funzione 'sfoca.m' sui valori da plottare, per attenuare i picchi. I grafici così ottenuti risultano decisamente più leggibili.

Si nota molto bene la curiosa forma che tende ad assumere il plot nel caso della PSF esponenziale. Per frequenze basse tende a 0, cioè le due matrici hanno autovalori simili; per frequenze di valori intermedi si crea una sorta di avvallamento centrale (il minimo è proprio sulla metà), e quindi il metodo con condizioni di Neumann ha autovalori maggiori; quando nella coppia di frequenze almeno una è molto alta, invece, il comportamento si ribalta ed è il metodo con condizioni periodiche ad avere autovalori maggiori.

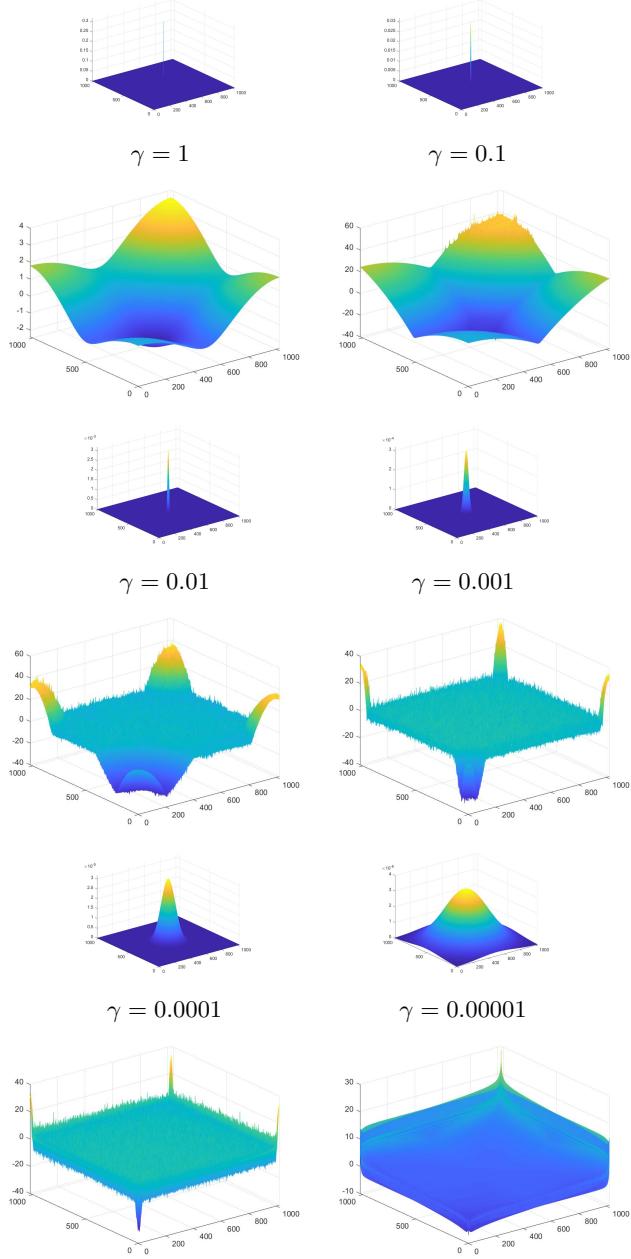


Figura 8: In alternativa, concentrandomi stavolta però solo sulla PSF esponenziale, ho pensato di migliorare la definizione aumentando sia le dimensioni del supporto della PSF sia le dimensioni richieste per la matrice di sfocatura da costruire, così da avere una maggiore definizione Variando inoltre il γ (= 0.1 nell'articolo) della gaussiana si nota come evolve il plot degli autovalori (sotto) al variare della dimensione relativa, all'immagine, della PSF (sopra).