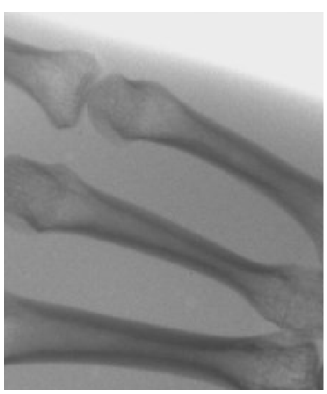
ENHANCEMENT – OPERATORI PUNTUALI

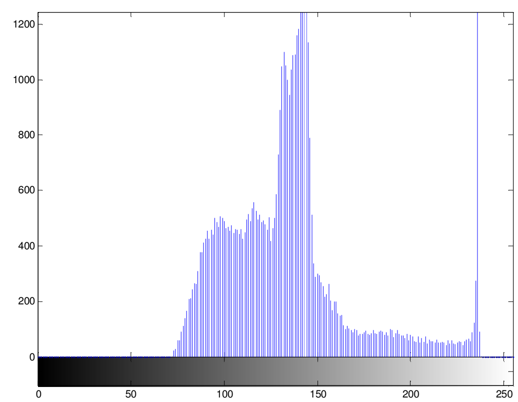
Operatori che migliorano la qualità dell'immagine assegnando al voxel (i,j) dell'immagine in uscita un valore dipendente dal solo valore del voxel (i,j) dell'immagine in ingresso.
La scelta del tipo di operatore puntuale può essere fatta valutando l'istogramma dell'immagine.

Funzione ***imhist.m***: fornisce l'istogramma della distribuzione dei livelli in un'immagine X.

- Immagini di tipo intensità
  imhist(X,N): istogramma su N livelli equispaziati di ampiezza A/(N-1) con A=realmax(class) o intmax(class); N se non specificato vale 256.
  QUINDI: prima di applicare imhist, le immagini di classe double devono essere scalate nel range [0,1].
- Immagini binarie
  imhist(X): istogramma su 2 livelli.
- Immagini di tipo indexed
  imhist(X,map): istogramma con un livello per ciascuna riga della map.

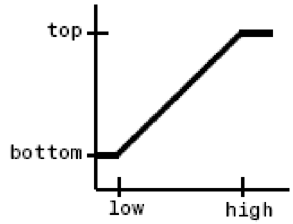
ESERCIZIO
Provare a ricavare l'istogramma dell'immagine contenuta nel file MANO.jpg, selezionando una regione di interesse comprendente sia lo sfondo che le strutture anatomiche. Provare a variare il numero di livelli dell'istogramma N=32, 64, 256.



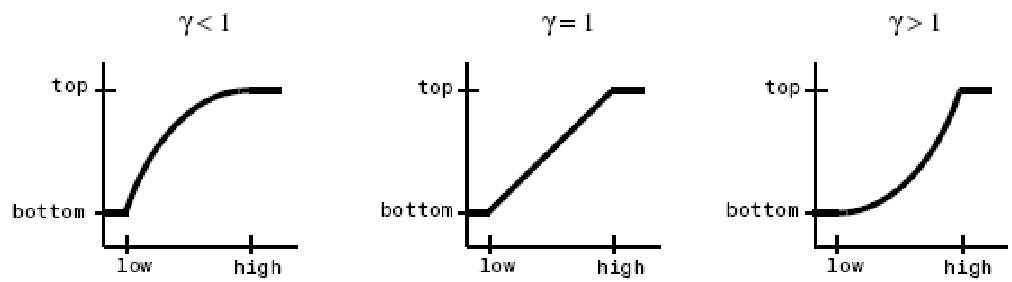


Verranno presi in considerazione due tipi di operatori puntuali:

1. **<u>Rimappatura dei livelli di intensità:</u>**
    - rimappatura lineare



    - rimappatura non lineare



Funzione ***imadjust.m***

- Immagini di tipo intensità

  Y=*imadjust(X,[LOW_in HIGH_in],[LOW_out HIGH_out], GAMMA)*;

  I valori dell'immagine X appartenenti all'intervallo [LOW_in HIGH_in] vengono rimappati nell'intervallo [LOW_out HIGH_out] con una trasformazione lineare se γ=1, non lineare altrimenti.

$$Y = \begin{cases} (HIGH\_out - LOW\_out) \cdot \left(\dfrac{X - LOW\_in}{HIGH\_in - LOW\_in} \right)^{\gamma} + LOW\_out & se \quad X \in [LOW\_in; HIGH\_in] \\ LOW\_out & se \quad X \leq LOW\_in \\ HIGH\_out & se \quad X \geq HIGH\_in \end{cases}$$

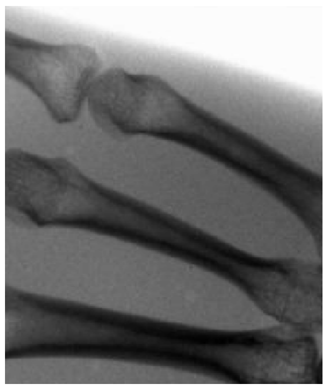
  imadjust(X) = imadjust(X,stretchlim(X, [0.01 0.99]),[0 1]): rimappa i livelli di intensità in modo che 1% di pixel sia saturato a 0 e 1% a 1.

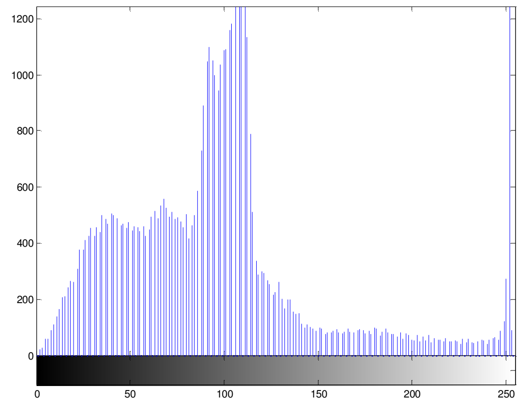
- Immagini di tipo indexed
  mapY=imadjust(mapX,…)
- Immagini RGB
  RGBY=imadjust(RGBX,…)

ESERCIZIO

Provare ad applicare la funzione imadjust.m alla porzione di mano.
Si può osservare come non ci siano pixel sotto il valore 72 o sopra il valore 238. Se l'immagine viene rimappata così da riempire i 256 valori di intensità disponibili, è possibile aumentare il contrasto:

Y=imadjust(X,[72/255 238/255],[0 1]);





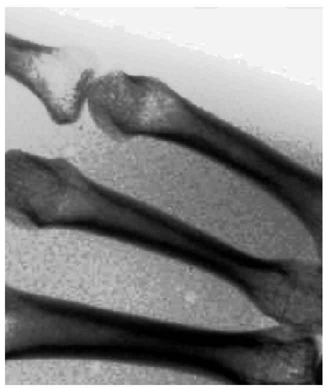
ESERCIZIO

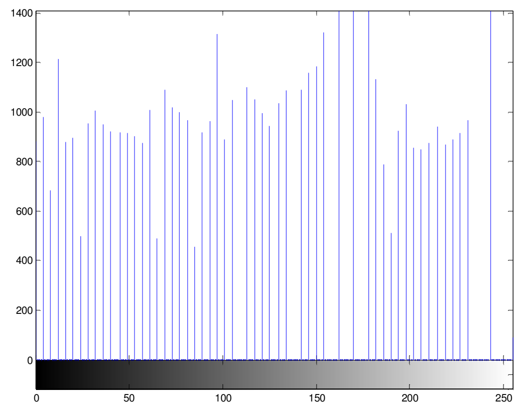
Provare a modificare l'istogramma delle immagini "disco1.jpg" e "disco2.jpg" al fine di meglio individuare i dettagli.

2. **<u>Equalizzazione dell'istogramma</u>:** ridistribuzione dei livelli di grigio dell'immagine in modo che l'istogramma sia il più possibile uniforme.

   Funzione ***histeq.m***
   Y=histeq(X,N) effettua automaticamente una equalizzazione dell'istogramma. Di default l'equalizzazione viene eseguita rispetto ad un istogramma uniforme su N=64 valori.

   <u>ESERCIZIO</u>
   Provare ad equalizzare l'istogramma delle tre immagini precedentemente considerate.





<u>ENHANCEMENT – OPERATORI LOCALI</u>

Operatori che migliorano la qualità dell'immagine assegnando al voxel (i,j) dell'immagine in uscita un valore dipendente dai valori dei voxel in un intorno del voxel (i,j) dell'immagine in ingresso.
Gli operatori locali vengono generalmente usati per ridurre il rumore sulle immagini.
Verranno presi in considerazione tre tipi di rumore:

- Rumore bianco Gaussiano additivo: rumore caratterizzato da un valore medio *M* e da una varianza D^2
  Per aggiungere ad un immagine X del rumore Gaussiano: *Y=**imnoise**(X,'gaussian',M,D^2);*

  Dato il valor medio *M* e la varianza D^2 (entrambi compresi tra 0 e 1), il 99.7% del rumore sarà compreso nell'intervallo di ampiezza 6*D* centrato attorno ad *M*. Nel caso in cui l'ampiezza 6*D* cada fuori dai valori ammessi, avrò saturazione del livello estremo corrispondente.
  Con questo tipo di rumore si possono simulare fonti sconosciute di rumore additivo. Infatti, per il teorema del limite centrale, la sommatoria di un gran numero di fonti di rumore indipendenti approssima un rumore gaussiano.

  <u>ESERCIZIO</u>
  Provare ad aggiungere del rumore bianco Gaussiano all'immagine della mano ed esaminare le variazioni dell'istogramma.
  g1=imnoise(X,'gaussian',0,0.01);
  g2=imnoise(X,'gaussian',0,0.1);

- Rumore sale&pepe: rumore impulsivo caratterizzato dalla frazione (in %) dell'immagine modificata.
  Per aggiungere ad un immagine X del rumore sale&pepe: *Y=**imnoise**(X,'salt & pepper',p);*

  Sostituisce p% (tra 0 e 1) dei pixel dell'immagine con pixel di intensità Gp (pepe) e p% con pixel di intensità Gs (sale).

  <u>ESERCIZIO</u>
  Provare ad aggiungere del rumore sale&pepe all'immagine della mano ed esaminare le variazioni dell'istogramma.
  sp1=imnoise(X,' salt & pepper',0.05);
  sp2=imnoise(X,' salt & pepper',0.2);

- Disturbo ad una fissata frequenza spaziale.

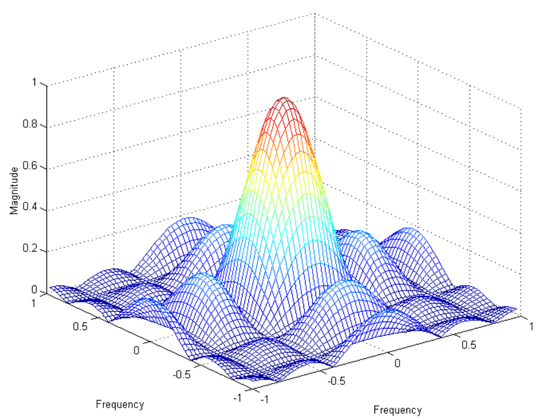
  <u>ESERCIZIO</u>
  Caricare il file noise3.mat, contenente un'immagine SPECT e la stessa immagine con sovrapposto del rumore sinusoidale ad alta frequenza. Visualizzare le immagini e il modulo della trasformata di Fourier.

A seconda della natura del rumore sovrapposto all'immagine, occorre scegliere la tecnica di rimozione. Se si conosce il tipo di rumore presente sull'immagine, la selezione del tipo di filtro da applicare sarà facilitata; se non lo si conosce, è possibile cercare di ricavare delle informazioni sul rumore esaminando una porzione omogenea dell'immagine (per esempio lo sfondo) e/o fare delle ipotesi sul tipo di rumore in base all'istogramma dei livelli di grigio.

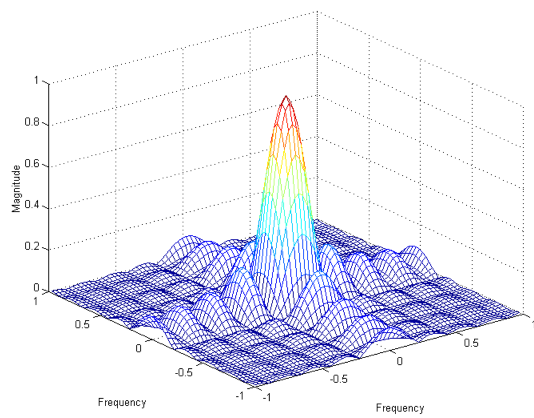
<u>FILTRAGGIO SPAZIALE LINEARE</u>

Il filtraggio spaziale lineare si basa sulla convoluzione 2-D dell'immagine X(i,j) con la risposta all'impulso del filtro h(i,j): tipicamente la maschera di h viene scelta di forma quadrata, di dimensioni dispari e simmetrica. Se la somma dei coefficienti del filtro h è pari ad 1, la luminosità dell'immagine sarà preservata.

- **Filtro medio**
  E' un passa basso; è il filtro migliore per rimuovere il rumore Gaussiano, ma la sua applicazione offusca l'immagine riducendo la visualizzazione dei contorni.
  h=fspecial('average',N) con N=dimensioni della maschera; la deviazione standard del rumore Gaussiano viene ridotta di un fattore $1/\sqrt{(N)}$.

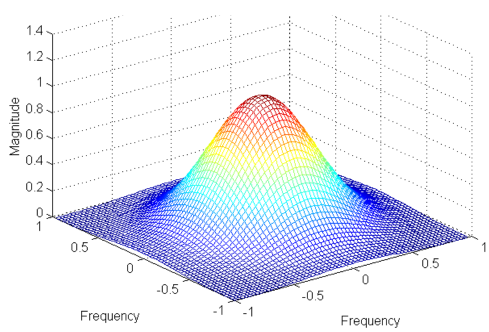


N=5



N=9

- **Filtro Gaussiano**
  E' anch'esso un filtro passa basso che quindi produce offuscamento. A differenza del filtro medio, ha simmetria circolare in frequenza.
  h=fspecial('gaussian',N,SIGMA); con N = dimensione della maschera
  SIGMA = dev.st. in pixel.



FILTRAGGIO SPAZIALE NON LINEARE

- **Filtro mediano**

  E' un filtro adatto ad eliminare il rumore sale e pepe, senza modificare i contorni. Sostituisce al pixel centrale il valore mediano dei pixel contenuti nella maschera considerata.

  Y=medfilt2(X,[M N]) con MxN dimensioni della maschera del filtro.

  ESERCIZIO

  Provare a filtrare le immagini rumorose create con filtri medio e mediano 3x3 e 5x5.

FILTRAGGIO SPAZIALE LINEARE IN FREQUENZA

Il filtraggio spaziale lineare può essere realizzato anche operando nelle frequenze: alla convoluzione nello spazio corrisponde infatti il prodotto nelle frequenze. Operare nelle frequenze consente di ridurre lo sforzo computazionale associato al filtraggio (il prodotto di due matrici costa meno di una convoluzione).

Si considerino il filtro lineare h(x,y), l'immagine f(x,y) e l'immagine filtrata g(x,y)=f(x,y)*h(x,y).
Come ottenere g(x,y) operando nelle frequenze?
Osservazioni:

- Il prodotto punto a punto tra la trasformata del filtro e la trasformata dell'immagine, G(n,m)= H(n,m) F(n,m), richiede che le dimensioni delle due trasformate siano coincidenti.
- In generale, H(n,m) ha valori complessi, così come F(n,m). L'operazione di moltiplicazione darà luogo a G(n,m) a valori complessi. Il filtraggio con una funzione complessa modifica però la fase dell'immagine originaria, che invece deve essere preservata durante le operazioni di filtraggio. Per fare in modo che questo non accada, H(n,m) deve essere reale, senza componente immaginaria. Filtri di tale tipo sono detti "zero-phase" e sono filtri simmetrici nello spazio rispetto all'origine.

ESERCIZIO

Provare a filtrare l'immagine spect_noise contenuta in noise3.mat, scegliendo il filtro più appropriato tra i filtri gaussiani, modificando i parametri che lo definiscono. Provare a filtrare l'immagine operando sia nello spazio che nelle frequenze.

```
h=fspecial('gaussian',17,2.5);
g=imfilter(spect_noise,h,'conv');

F=fft2(spect_noise,91,91);
h1=fspecial('gaussian',91,2.5);
hc=ifftshift(h1);
H=fft2(hc);
G=F.*H;
g1=ifft2(G);
g1=g1(1:end-1,1:end-9);
```