

Introduzione alle tecniche di ricostruzione iterativa per la tomografia ad emissione

Focus principale sugli algoritmi ML-EM e OS-EM



UNIVERSITÀ DI PISA

Michele Scipioni

Ph.D. Student

Dip. Ingegneria dell'informazione

Università di Pisa

Corso di Immagini Biomediche, 28 Ottobre 2016

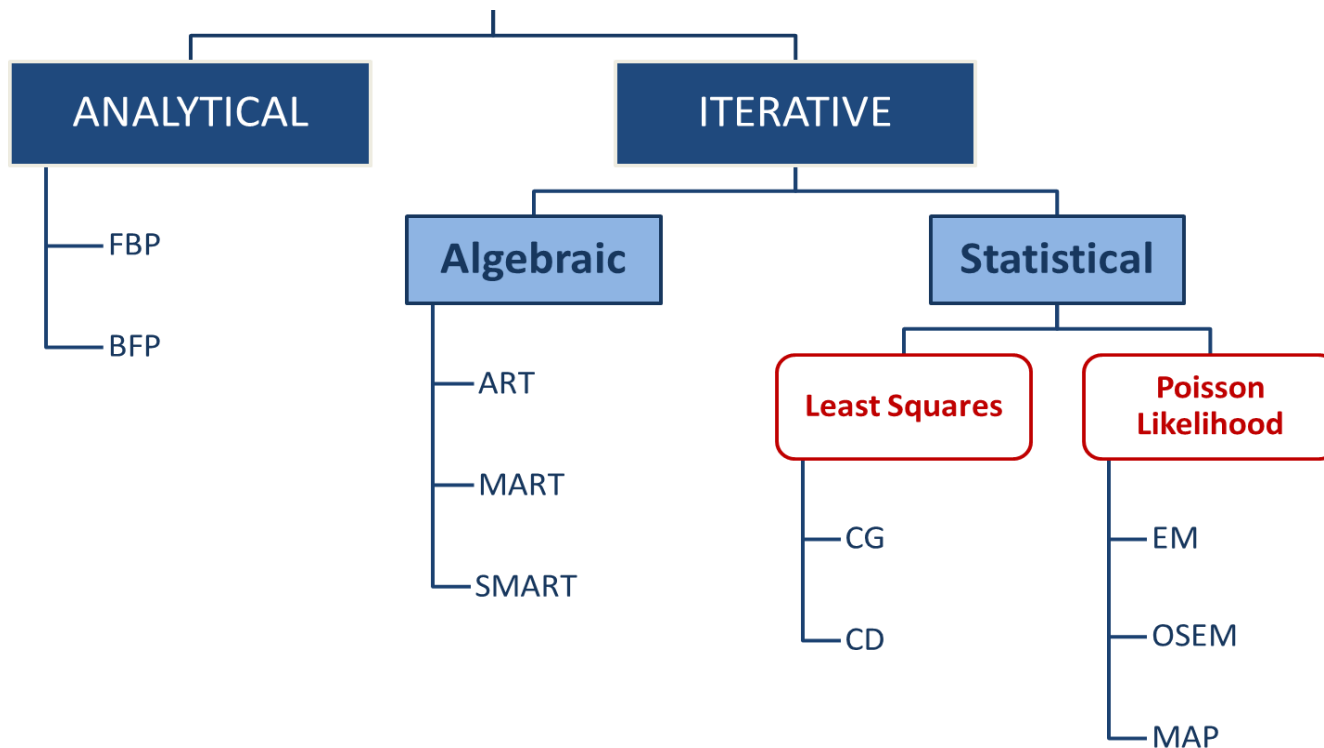


Outline

- *Introduzione*
- *Ricostruzione tomografica: un problema lineare inverso*
- *Elementi di un algoritmo iterativo*
- *Criteri di ottimizzazione*
- *Algoritmi di ricostruzione iterativa*
- *Esercitazione*



Reconstruction Methods



Metodi analitici vs Metodi iterativi

Metodi analitici (FBP)

*Problema matematico basato sull'**inversione della trasformata di Radon discreta**. I dati sono integrali di linea e non si tenta di modellare esplicitamente la randomicità tipica del processo di conteggio dei fotoni.*

Metodi iterativi

Generico modello lineare che consente una descrizione dettagliata dei meccanismi di blurring ed attenuazione. Oltre a questo, le tecniche statistiche di ricostruzione cercano di incorporare nel modello anche una descrizione probabilistica del rumore.



Metodi analitici vs Metodi iterativi

Trade-off

Il trade-off tra tecniche iterative ed FBP è quindi una scelta tra accuratezza ed efficienza della ricostruzione.

Tecniche analitiche

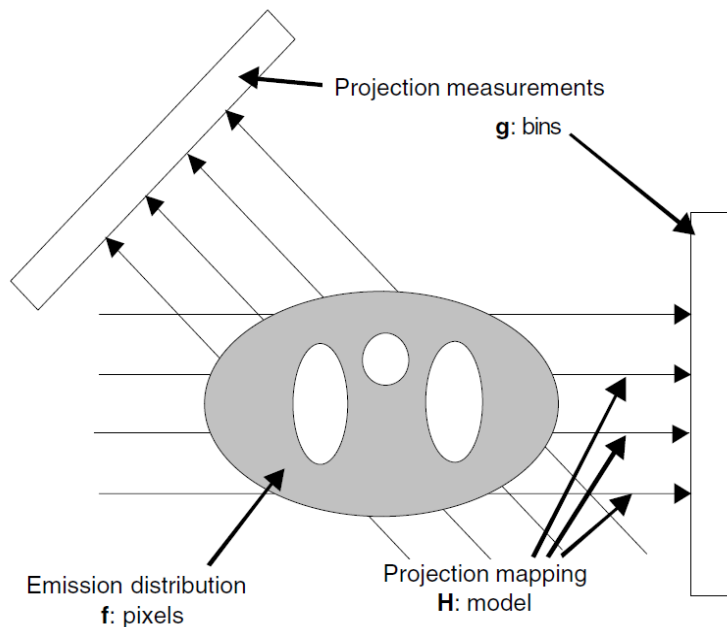
- Modello non fedele alla natura dei dati reali
- Immagini ottenute velocemente ma poco accurate
- Immagini molto rumorose

Tecniche iterative/statistiche

- Modello matematico molto più complesso
- Non esiste soluzione analitica in forma chiusa
- Tempi di calcolo molto maggiori

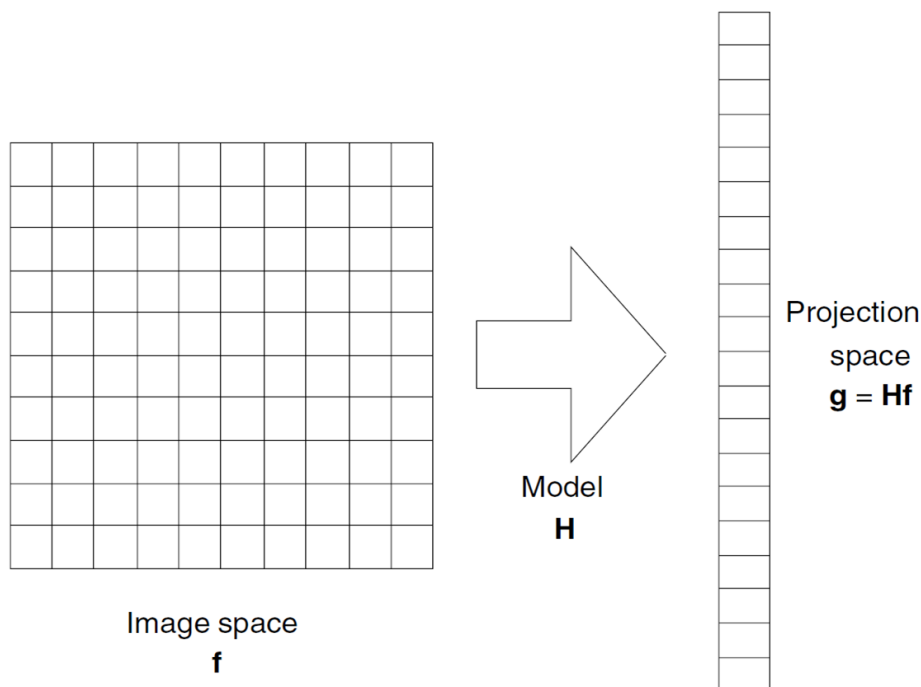


Modello lineare



- \mathbf{g} è l'insieme di proiezioni angolari misurate, il sinogramma
- \mathbf{H} è la matrice di sistema, e descrive l'intero processo di registrazione delle immagini. Ogni elemento di \mathbf{H} rappresenta il contributo medio di un pixel dell'oggetto al valore di unbin del sinogramma.
- Ogni elemento di \mathbf{f} rappresenta un pixel /voxel dello spazio immagine (indifferentemente una fetta 2D o un volume 3D)

Modello lineare



$$g_i = \int_{\mathbb{R}^D} f(x) h_i(x) dx, i = 1, \dots, P$$

$$g_i = \mathbf{h}_i(x)^T \mathbf{f}, i = 1, \dots, P$$

$$\mathbf{g} = \mathbf{H}\mathbf{f}$$

$$\mathbf{f} = \mathbf{H}^{-1}\mathbf{g}$$

H è grandissima, non quadrata e non invertibile

Modello statistico

Metodi algebrici: interessati al **comportamento medio** del sistema

Metodi statistici: le misure g_i sono trattate come **variabili aleatorie** descritte da distribuzione di probabilità di **Poisson** (distribuzione tipica di fenomeni di conteggio)

Una riformulazione più corretta della dovrebbe infatti essere:

$$E[\mathbf{g}] = \bar{g}_i = \sum_j h_{ij} f_j = \mathbf{H}_i \mathbf{f}$$

Valore atteso (medio)

$$p(\mathbf{g}|\mathbf{f}) = \prod_i p(g_i|\mathbf{f}) = \prod_i \frac{e^{-\mathbf{H}_i \mathbf{f}} (\mathbf{H}_i \mathbf{f})^{g_i}}{g_i!}$$

Probabilità di misurare \mathbf{g} , data un'immagine vera \mathbf{f}



Modello statistico

Nota bene

Il modello di Poisson descrive bene i dati di emissione non corretti ed è il più utilizzato nel campo della tomografia ad emissione (PET e SPECT).

Se il sistema di imaging corregge internamente i sinogrammi per le coincidenze random (sottraendo una stima degli stessi dai conteggi totali), la statistica muta.

Solitamente si dovrebbe ricorrere ad un modello **Shifted Poisson**, ma più generalmente si opta per un **modello Gaussiano** più semplice

se il numero medio di eventi è sufficientemente ampio, la distribuzione di Poisson può essere ben approssimata da una gaussiana con media e varianza uguali tra loro e di valore \bar{g}_i



Elementi di un algoritmo iterativo

Ogni metodo di ricostruzione iterativo si compone necessariamente di due elementi fondamentali:

- **criterio di ottimizzazione:** è il criterio rispetto al quale è possibile determinare quale immagine deve essere considerata la stima migliore dell'immagine vera;
- **algoritmo di ottimizzazione:** tecnica computazionale finalizzata a cercare la soluzione richiesta dal criterio di ottimizzazione.

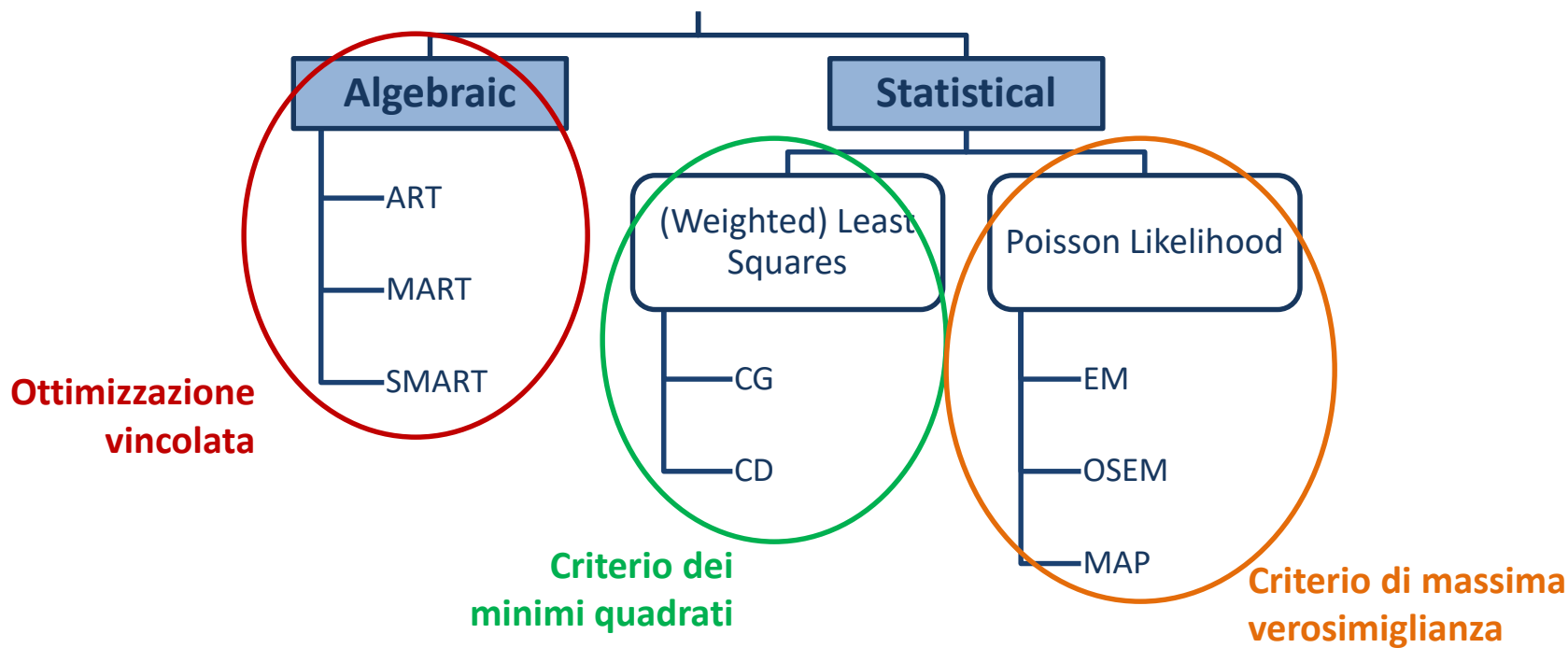
Detto in breve ...

Il criterio e la strategia di ricostruzione, l'algoritmo la definizione dei singoli passi necessari ad implementare tale strategia.



Criteri di ottimizzazione

ITERATIVE METHODS



Ottimizzazione vincolata (ART)

Criterio di ottimizzazione vincolata

*Ricerca di un'immagine che soddisfi una serie di **vincoli imposti dai dati misurati** e da alcune **ipotesi a priori** (ad esempio la non-negatività dei pixel). Questa via ha portato alla definizione di una serie di algoritmi che ricadono nella categoria delle **tecniche algebriche di ricostruzione** (ART).*

Il **punto debole** di questi approcci è che non offrono nessun meccanismo che consenta di incorporare un modello statistico esplicito dei dati con cui abbiamo a che fare.



Criterio dei minimi quadrati (LS, WLS)

Criterio LS

Scegliere il valore di \mathbf{f} che, se osservato attraverso la matrice di sistema \mathbf{H} , garantirebbe delle proiezioni $\mathbf{H}\mathbf{f}$ più possibile simili alle proiezioni osservate \mathbf{g} (distanza Euclidea).

$$\hat{\mathbf{f}} = \arg \min_f (\mathbf{g} - \mathbf{H}\mathbf{f})^T \mathbf{D} (\mathbf{g} - \mathbf{H}\mathbf{f})$$

$$= \arg \min_f \sum_{i=1}^p d_i \left(g_i - \sum_{j=1}^N h_{ij} f_j \right)^2$$



$$\hat{\mathbf{f}} = (\mathbf{H}^T \mathbf{D} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{D}^{-1} \mathbf{g}$$

(SOLUZIONE IN FORMA CHIUSA)

\mathbf{D} è una matrice diagonale i cui elementi d_i sono $(\text{var}[g_i])^{-1}$. Nella tomografia ad emissione, i cui dati sono distribuiti secondo Poisson, la varianza è uguale alla media, per cui $d_i = g_i^{-1}$

Questa soluzione non è usata spesso nella tomografia ad emissione per le grandi dimensioni della matrice di sistema \mathbf{H} , e spesso si preferisce **ricorrere a soluzioni iterative**.



Criterio di massima verosimiglianza

Criterio ML

Scegliere come ricostruzione ottima, $\hat{\mathbf{f}}$, quella che garantisce la massima probabilità di aver prodotto i dati misurati (massimizza la funzione di verosimiglianza $p(\mathbf{g}|\mathbf{f})$ o $L(\mathbf{f})$).

$$\hat{\mathbf{f}} = \arg \max_f p(\mathbf{g}|\mathbf{f})$$

Il criterio ML cerca la soluzione caratterizzata dalla **maggiore vicinanza statistica alle osservazione misurate**.



Criterio di massima verosimiglianza

Proprietà dello stimatore ML

- è **asintoticamente non polarizzato** (*asymptotically unbiased*), ossia al crescere del numero di osservazioni, la stima tende al valore vero $\left(E[\hat{f}] \rightarrow f\right)$
- è **asintoticamente efficiente**, ossia è lo stimatore a varianza minima, tra tutti gli stimatori nonpolarizzati. Detto altrimenti, lo stimatore ML è meno suscettibile al rumore presente nei dati.



Limite intrinseco degli approcci ML, LS e WLS

Sfortunatamente per noi, nonostante la varianza (rumore) delle immagini ET ricostruite con stimatore ML sia la più bassa ottenibile con uno stimatore non polarizzato, questa continua ad essere ancora inaccettabilmente alta. Questo segue direttamente dalla loro tendenza a garantire la **massima consistenza con i dati misurati**: dato che *i dati sono intrinsecamente rumorosi*, l'immagine che è massimamente consistente con essi sarà per forza di cose rumorosa a sua volta.



Questo vale anche per stimatore LS e WLS!

(caso LS)

$$\begin{aligned}\hat{\mathbf{f}} &= \mathbf{H}^+ \mathbf{g} = \mathbf{H}^+ (\mathbf{H} \mathbf{f} + \mathbf{n}) \\ &= \mathbf{f} + \mathbf{H}^+ \mathbf{n}\end{aligned}$$

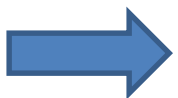
- Ipotesi di rumore additivo
- H nella tomografia ad emissione agisce come sistema passa-basso
- La sua inversa H^+ di conseguenza è passa alto

Possibili soluzioni

La soluzione consiste nell'accettare un certo *bias* nelle immagini ricostruite in cambio di una riduzione di varianza. Per far questo si introduce uno *smoothing* spaziale nelle immagini, che ovviamente va a ridurre il rumore a discapito della vicinanza della stima al valore vero.

Lo *smoothing* può essere introdotto in due modi:

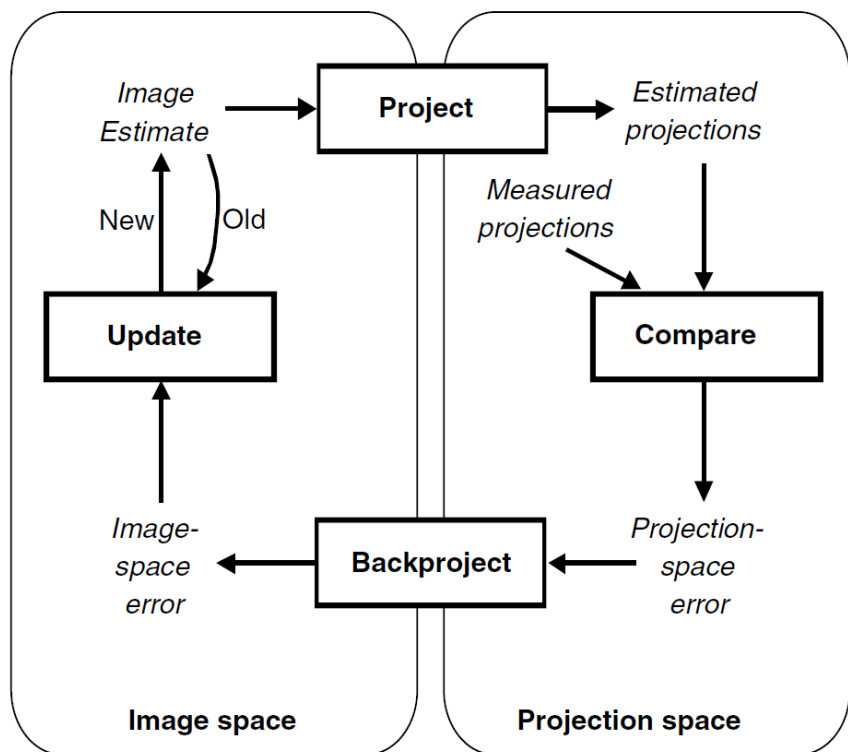
- **esplicitamente**: applicando un filtraggio passa basso all'immagine ricostruita
- **implicitamente**: interrompendo prematuramente (prima di convergere all'effettiva soluzione ML) le iterazioni dell'algoritmo



Approcci Bayesiani di stima regolarizzata (Maximum A Posteriori)

$$\hat{\mathbf{f}} = \arg \max_f [\ln p(\mathbf{g}|\mathbf{f}) + \ln p(\mathbf{f})]$$

Struttura generale di un algoritmo iterativo



$\hat{f}^{(n)}$ stima corrente all'iterazione n

$\hat{f}^{(n)} \rightarrow \hat{g}^{(n)}$ la stima corrente viene proiettata nel dominio dei sinogrammi

$\hat{g}^{(n)} \Leftrightarrow g$ la proiezione predetta viene confrontata con i valori misurati g

e_g vettore di errore definito nello spazio proiettato

e_f vettore di errore nello spazio immagine attraverso retroproiezione

$\hat{f}^{(n+1)}$ stima aggiornata sulla base del feedback di errore

Derivazione ML-EM: *missing data*

L'approccio EM è standard per la soluzione di problemi ML in cui non si hanno a disposizione *tutti* i dati che renderebbero semplice la soluzione (**problema dei dati mancanti**)



Astrazione

ipotizziamo l'esistenza di un **set completo di dati S** in cui conosciamo sia il numero di fotoni misurati nel bin j , sia quanti, di questi, provengono dal pixel k

E-step

$$Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)}) = E[\ln p(\mathbf{s}|\mathbf{f})|\mathbf{g}, \hat{\mathbf{f}}^{(n)}]$$

→ *Aspettazione della log-likelihood del set completo*

M-step

$$\hat{\mathbf{f}}^{(n+1)} = \operatorname{argmax}_{\mathbf{f}} Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)})$$

→ *Massimizzazione della stessa*



Derivazione ML-EM: *E-step*

s_{im} = elemento del set completo di dati (*variabile aleatoria Poissoniana*)

$$g_i = \sum_m s_{im}$$

$$E[s_{im}] = h_{im} f_m$$

$$p(\mathbf{s}|\mathbf{f}) = \prod_i \prod_m \frac{E[s_{im}]^{s_{im}} e^{-E[s_{im}]}}{s_{im}!}$$

$$\ln p(\mathbf{s}|\mathbf{f}) = \sum_i \sum_m [s_{im} \ln(h_{im} f_m) - h_{im} f_m - \ln(s_{im}!)]$$

$$Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)}) = E[\ln p(\mathbf{s}|\mathbf{f})|\mathbf{g}, \hat{\mathbf{f}}^{(n)}]$$

E-step

$$= \sum_i \sum_m \{E[s_{im} | \mathbf{g}, \hat{\mathbf{f}}^{(n)}] \ln(h_{im} f_m) - h_{im} f_m - E[\ln(s_{im}!)]\}$$



Derivazione ML-EM: *M*-step

E-step

$$Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)}) = \sum_i \sum_m \{E[s_{im} | \mathbf{g}, \hat{\mathbf{f}}^{(n)}] \ln(h_{im} f_m) - h_{im} f_m - E[\ln(s_{im}!)]\}$$

$$E[s_{im} | \mathbf{g}, \hat{\mathbf{f}}^{(n)}] = g_i \frac{h_{im} \hat{f}_m}{\sum_m h_{im} \hat{f}_m} \stackrel{\text{def}}{=} p_{im}$$

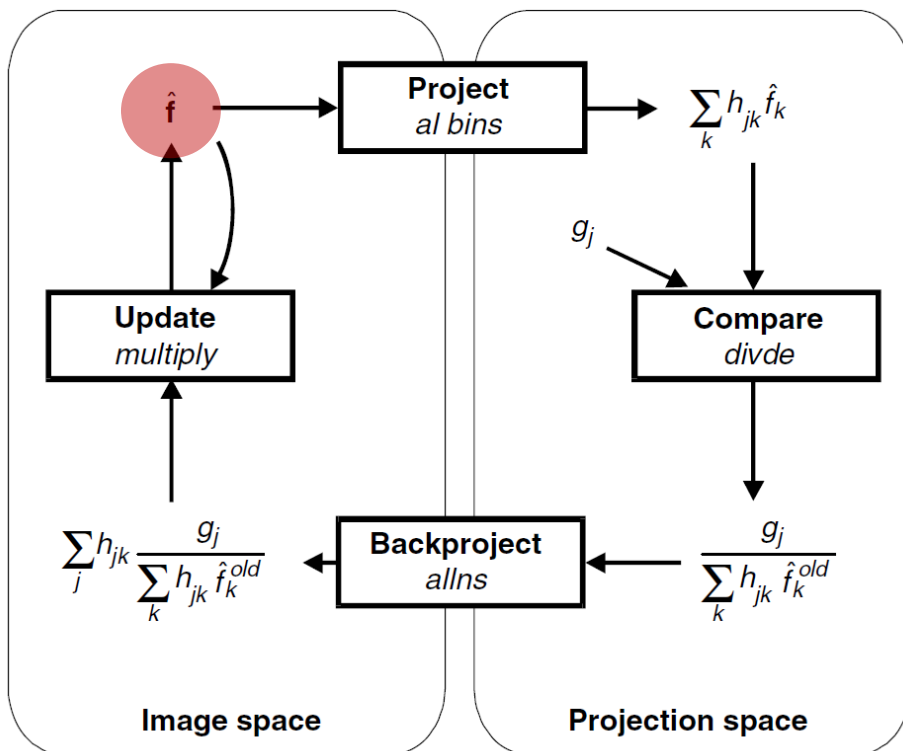
$$\frac{\partial Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)})}{\partial f_j} = 0 = \sum_i \frac{p_{ij}}{\hat{f}_j^{(n+1)}} - h_{ij} \quad \xrightarrow{\text{yields}}$$

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}$$

M-step



Algoritmo ML-EM



1) Stima iniziale uniforme dell'immagine

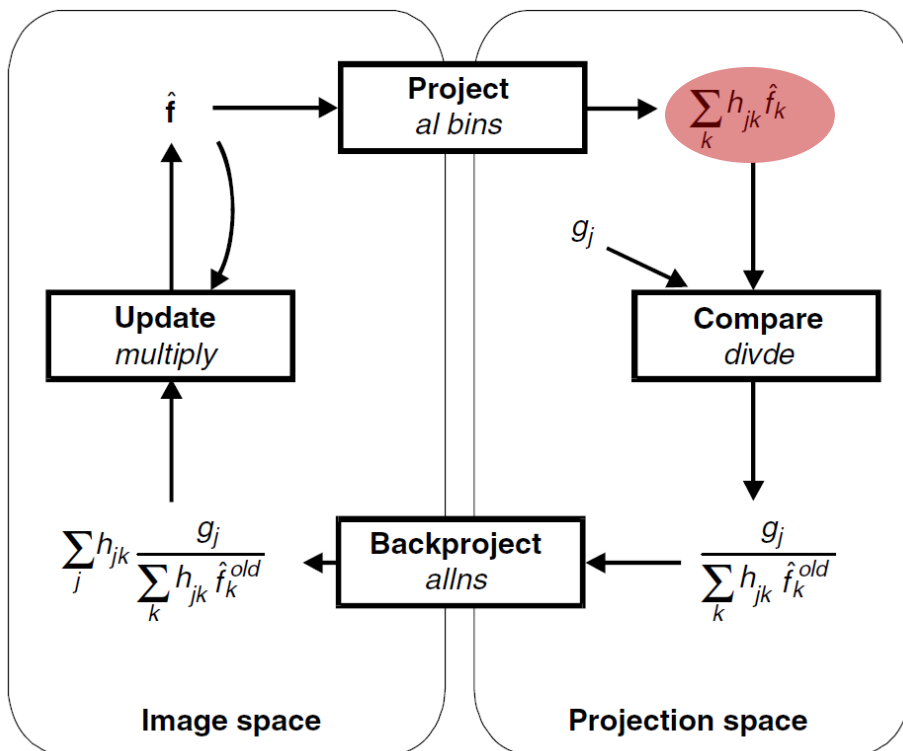
$$\hat{f}_j^{(0)}$$

ML-EM

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}$$



Algoritmo ML-EM



2) Simulazione della misura a partire dalla stima (*proiezione*)

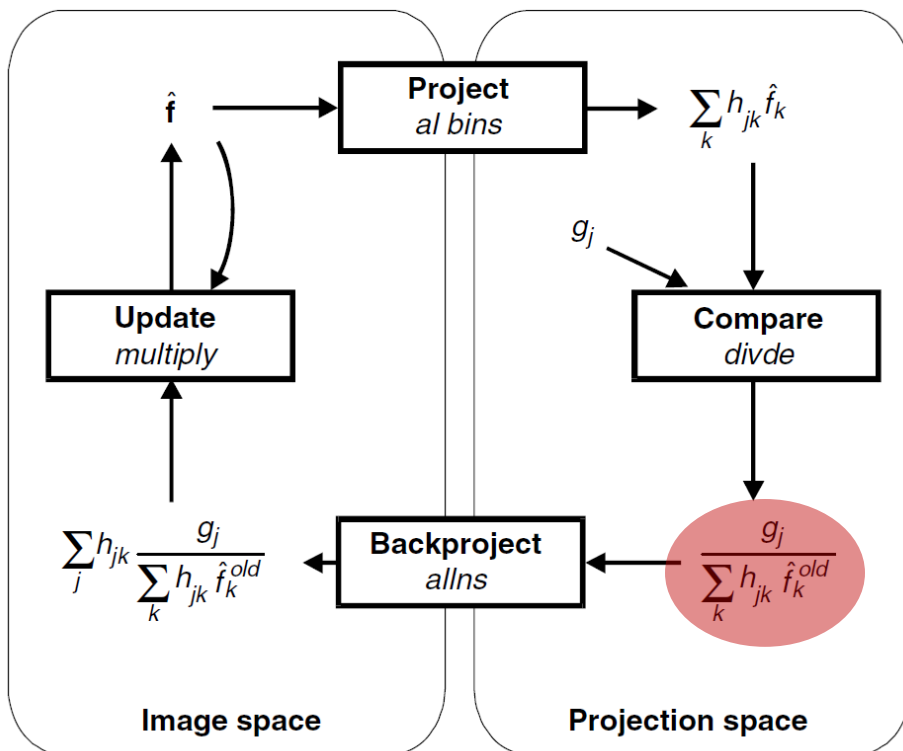
$$\hat{g}_i^{pred} = \sum_k h_{ik} \hat{f}_k^{(n)}$$

ML-EM

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}$$



Algoritmo ML-EM



3) Confronto con la misura vera

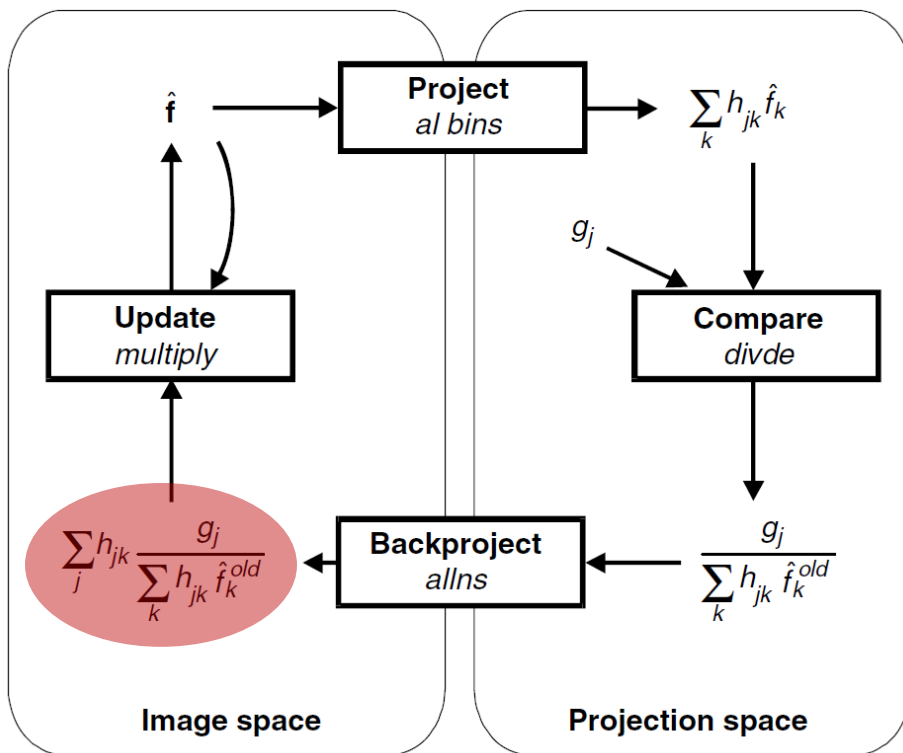
$$e_i(g) = \frac{g_i}{\hat{g}_i^{pred}}$$

ML-EM

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}$$



Algoritmo ML-EM



4) Passaggio al dominio immagine (backprojection)

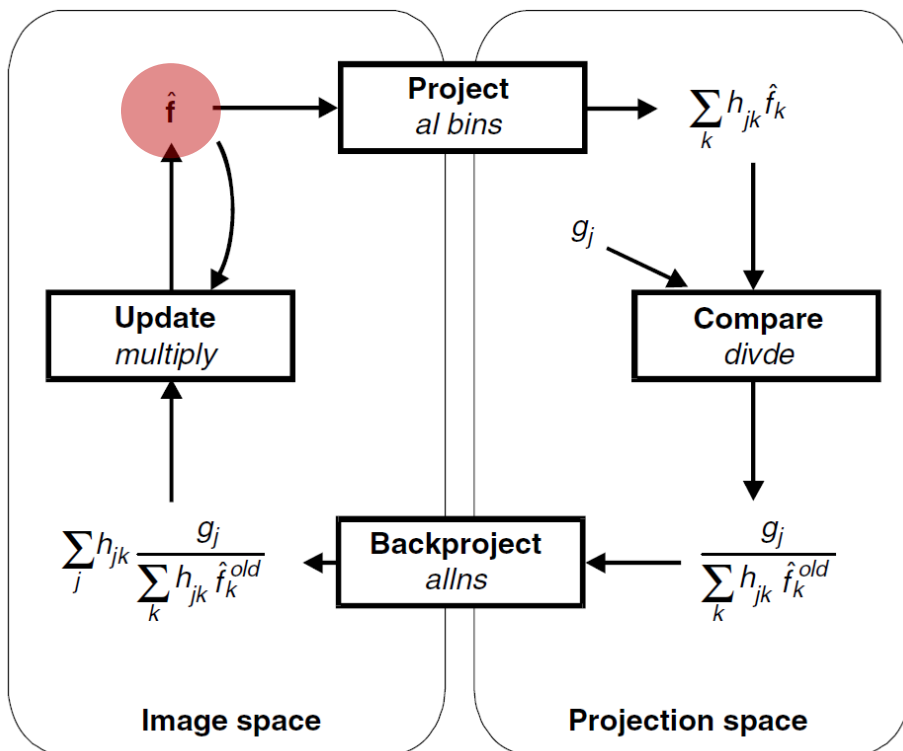
$$e_i^{(f)} = \sum_j h_{ij} e_j^{(g)}$$

ML-EM

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}$$



Algoritmo ML-EM



5) Aggiornamento della stima (pesatura della stima precedente)

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_i h_{ij}} e_i(f)$$

ML-EM

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}$$



Algoritmo ML-EM: *numero di iterazioni*

1) Convergenza garantita e predicibile ma lenta

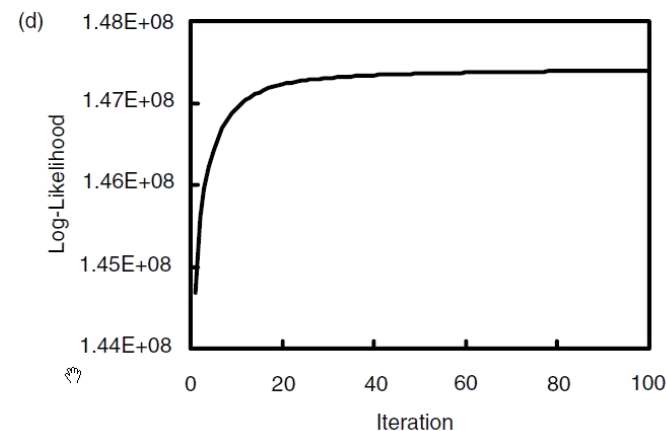
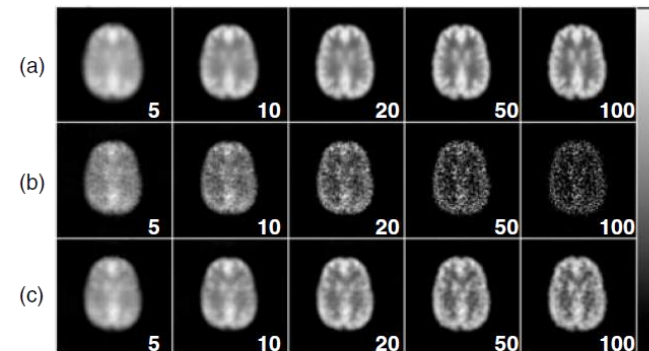
- Dati rumorosi: almeno 30, 50 iterazioni necessarie
- Tempi di esecuzione circa 2 ordini di grandezza superiori alla FBP

2) Immagini finali rumorose

- Tende alla massima consistenza con misure rumorose ...

Nota bene ...

La ML-EM ricostruisce prima le **frequenze spaziali più basse**, per poi sviluppare gradualmente le componenti ad alta frequenza: bloccare le iterazioni equivale quindi implicitamente ad un **filtraggio passa-basso** dell'immagine ricostruita.



Algoritmo OS-EM: *accelerare la convergenza*

$$\underbrace{\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}}_{\text{ML-EM}} \Rightarrow \underbrace{\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i' \in S_n} h_{i'j}} \sum_{i \in S_n} h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}}_{\text{OS-EM}}$$

La retroproiezione viene applicata soltanto ai bin che appartengono al sotto insieme (*subset*) S_n del sinogramma. Ad ogni aggiornamento, un diverso subset viene preso in considerazione in quella che viene definita *sotto-iterazione* dell'algoritmo (mentre un intero passaggio di tutti i subset costituisce un'iterazione vera e propria):

- $N_{\text{subset}} = 1$ \rightarrow ci riconduciamo al caso ML-EM
- $N_{\text{subset}} = N_{\text{proiezioni}}$ \rightarrow otteniamo un algoritmo simile alla (M)ART



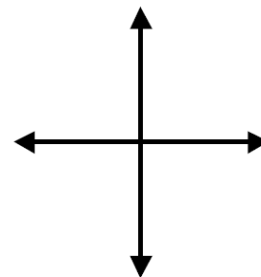
Algoritmo OS-EM: *organizzazione dei subsets*

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i' \in S_n} h_{i'j}} \sum_{i \in S_n} h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}$$

PRINCIPALI REGOLE DI SCELTA

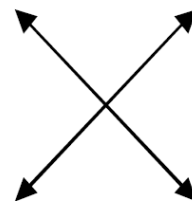
- In ogni subset deve comparire il contributo di tutti i pixel del FOV
- Le proiezioni che appartengono ad un subset sono scelte in modo da avere massima distanza angolare possibile tra loro
- Anche l'alternanza nel processing dei subset cerca di massimizzare la differenza tra subset successivi

(a)



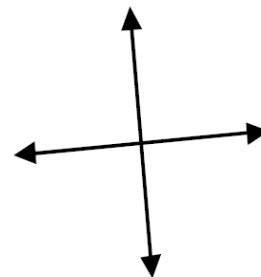
Subset 1
Apply ML-EM for
these four views.

(b)



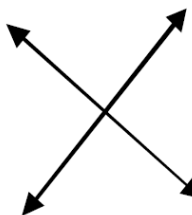
Subset 2
Then, for
these four.

(c)



Subset 3
Next, these

(d)



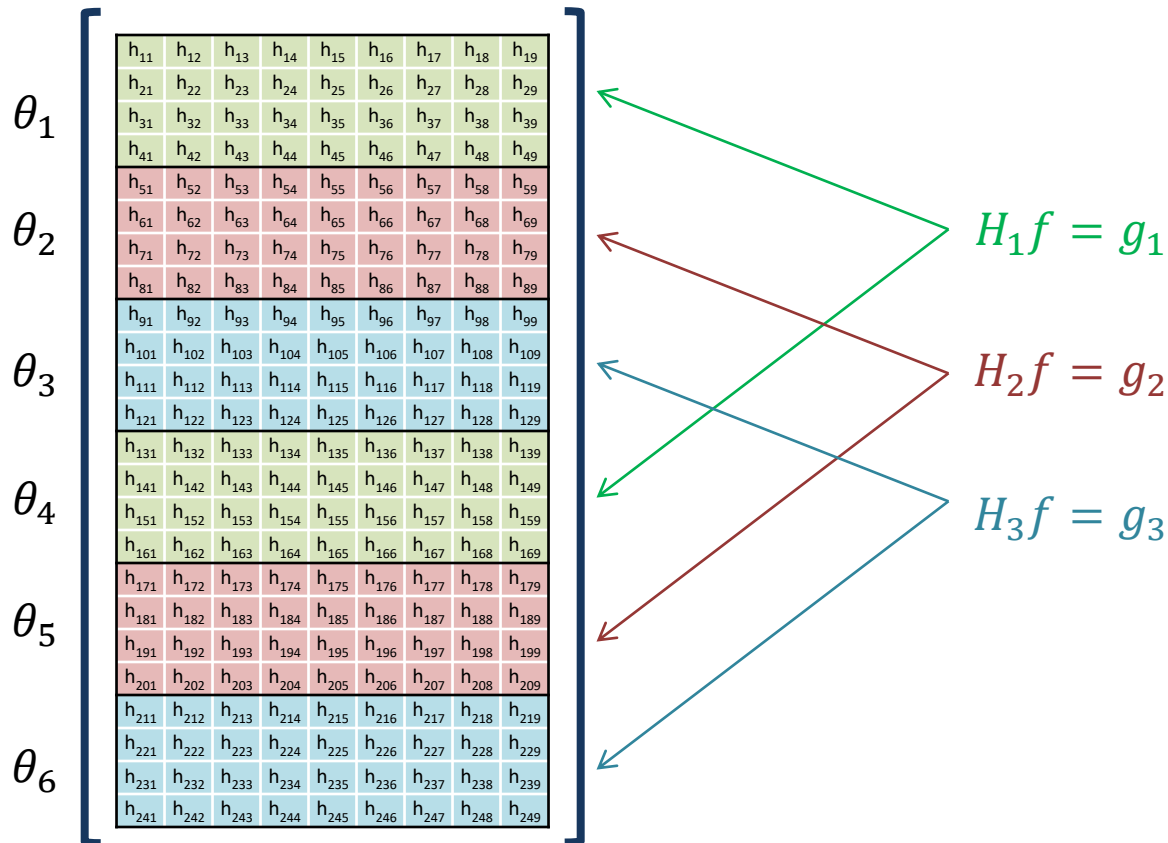
Subset 4
And, so on until

Algoritmo OS-EM: *organizzazione dei subsets*

$$f \xrightarrow{3 \times 3} \begin{bmatrix} f_1 & f_4 & f_7 \\ f_2 & f_5 & f_8 \\ f_3 & f_6 & f_9 \end{bmatrix}$$

 $H_{24 \times 9}$

$$\begin{aligned} N_{\text{proiezioni}} &: 6 \\ N_{\text{posizioni_angolari}} &: 4 \\ g &\rightarrow [6 \times 4] \end{aligned}$$



OS-EM vs ML-EM

- Il numero di subset scelto determina il **grado di accelerazione** della OS-EM
- (Empiricamente) dopo N iterazioni di OS-EM si raggiunge un **punto di convergenza** paragonabile a quello ottenuto con $(N_{subset} \times N)$ iterazioni di ML-EM
- Il **rumore cresce a velocità comparabile** e l'algoritmo va arrestato (ed eventualmente filtrato) ancora più precocemente
- Nonostante la somiglianza con la ML-EM, la OS-EM **non è realmente un algoritmo EM** e quindi non c'è prova teorica della sua convergenza



Correzioni iterative (ML-EM e OS-EM)

Abbiamo detto precedentemente che se il sistema di imaging corregge internamente i sinogrammi prima che avvenga la ricostruzione, la statistica muta e non sarebbe più valido lavorare nell'ipotesi di Poissonianità dei dati su cui si basano ML-EM ed OS-EM.

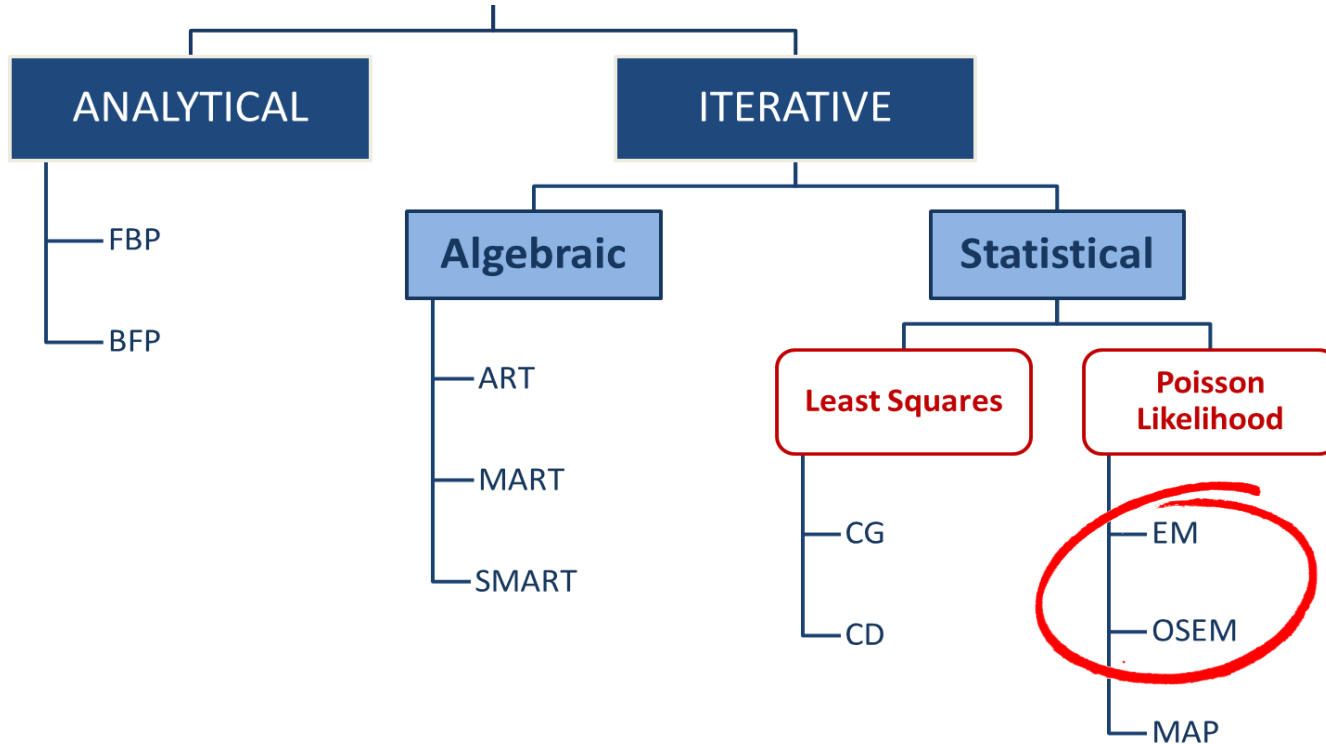
Tra i vantaggi del meccanismo di *feedback* che è alla base di queste tecniche iterative c'è però anche la possibilità di **incorporare delle stime delle sorgenti di disturbo** (le stesse che in maniera inversa verrebbero usate per pre-correggere i sinogrammi) **all'interno dello step di proiezione**, riuscendo quindi a ricostruire e correggere le immagini allo stesso tempo.

$$\hat{f}^{(n+1)} = \frac{\hat{f}^{(n)}}{H^T \mathbf{1}} H^T \frac{g}{\text{att.*} (H \hat{f}^{(n)}) + \text{scatt} + \text{rand}} \quad \left. \vphantom{\frac{g}{\text{att.*} (H \hat{f}^{(n)}) + \text{scatt} + \text{rand}}} \right\} \text{ML-EM}$$

$$\text{OS-EM} \left\{ \begin{aligned} \hat{f}_k^{(n+1)} &= \frac{\hat{f}_k^{(n)}}{H_k^T \mathbf{1}_k} H_k^T \frac{g_k}{\text{att}_{k.*} (H_k \hat{f}_k^{(n)}) + \text{scatt}_k + \text{rand}_k} \end{aligned} \right.$$



Reconstruction Methods

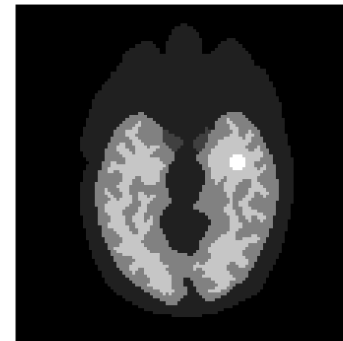


Esercitazione (0)

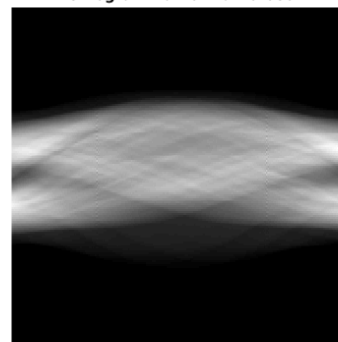
MATERIALE

- Fantoccio cerebrale 2D (111x111 pixel) preparato (file ***brain.mat***)
- Funzione MATLAB ***Calcolo_A.m*** già utilizzata nell'esercitazione sulle tecniche di ricostruzione analitiche
- Scheletro dell'esercitazione da svolgere
- Scheletro della funzione ***Calcolo_Ablock.m*** con descrizione di quale debba essere la sua struttura e il suo funzionamento.

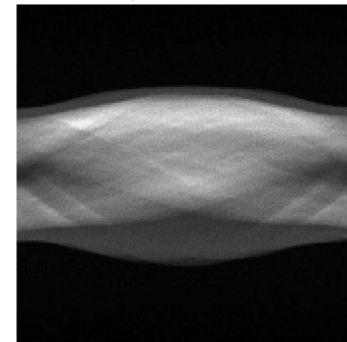
Fantoccio



Sinogramma non rumoroso



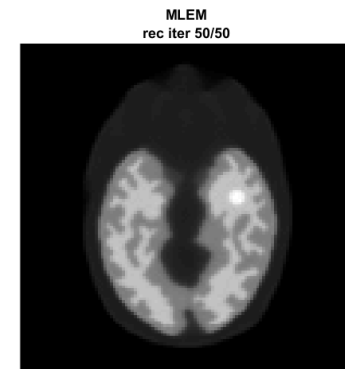
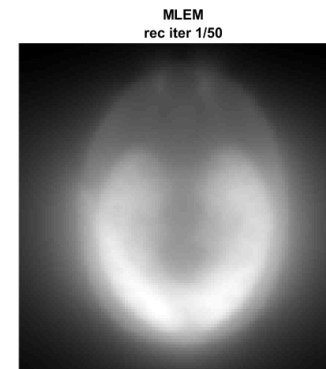
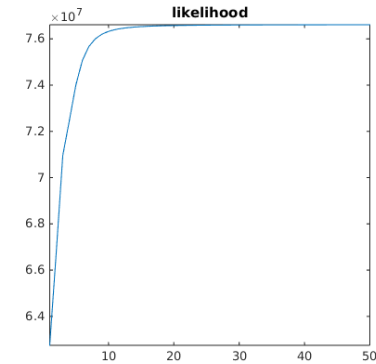
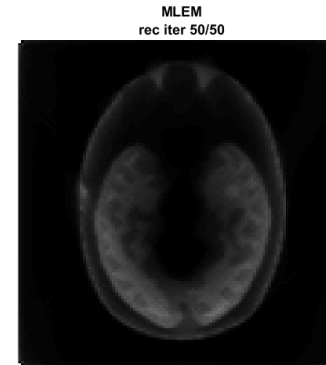
Sinogramma rumoroso



Esercitazione (1)

1. Ricostruzione ML-EM

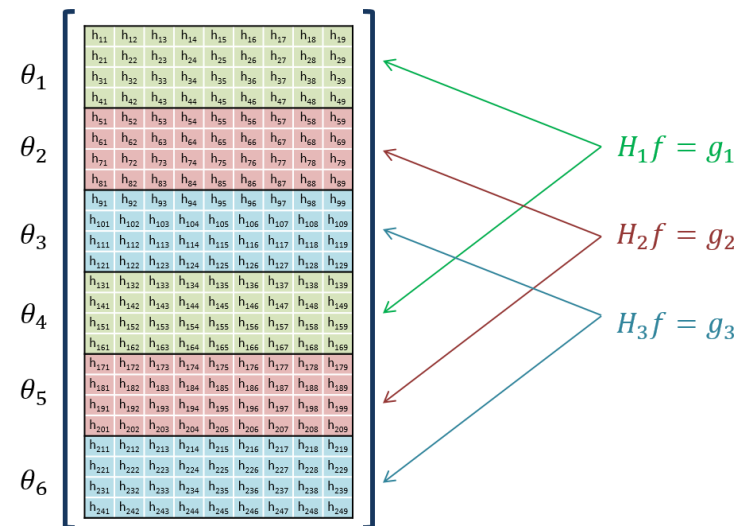
- Implementare l'algoritmo ML-EM per la ricostruzione del sinogramma rumoroso generato ai punti precedenti.
- Valutare la qualità della ricostruzione a seconda che vengano corretti o meno i disturbi simulati.
- Salvare l'immagine intermedia ricostruita ad ogni iterazione in un vettore 3D (N,N,iter mlem)



Esercitazione (2)

2. Ricostruzione OSEM: calcolo dei blocchi della matrice di sistema

- Creare una funzione esterna (partire dal le Calcolo Ablock.m fornito) per l'estrazione dei blocchi della matrice di sistema con cui ricostruire i singoli subset del sinogramma.
- La descrizione della funzione è fornita nel le dedicato: è importante assicurarsi che restituisca in output 'nblock' segmenti della matrice di sistema A e, per ciascuno di essi, tenga traccia delle proiezioni che fanno parte del subset a cui è associato un determinato blocco.

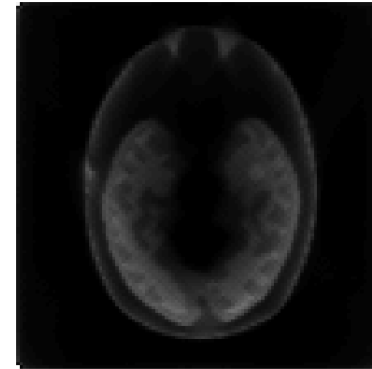


Esercitazione (3)

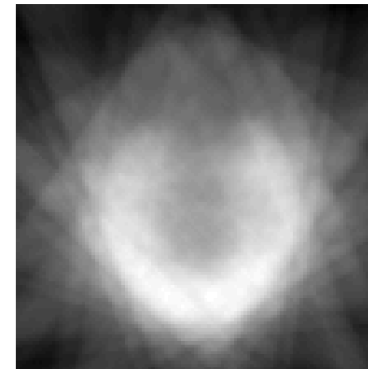
1. Ricostruzione OS-EM

- Implementare l'algoritmo OS-EM per la ricostruzione del sinogramma rumoroso generato ai punti precedenti.
- Valutare la qualità della ricostruzione a seconda che vengano corretti o meno i disturbi simulati.
- Salvare l'immagine intermedia ricostruita ad ogni iterazione in un vettore 3D ($N, N, \text{iter_mlem} * \text{nblock}$)

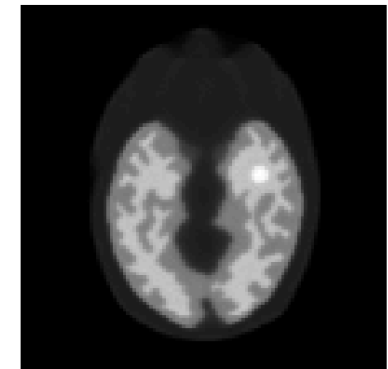
OSEM
rec iter 2/2
subset 25/25



OSEM
rec iter 1/2
subset 1/25



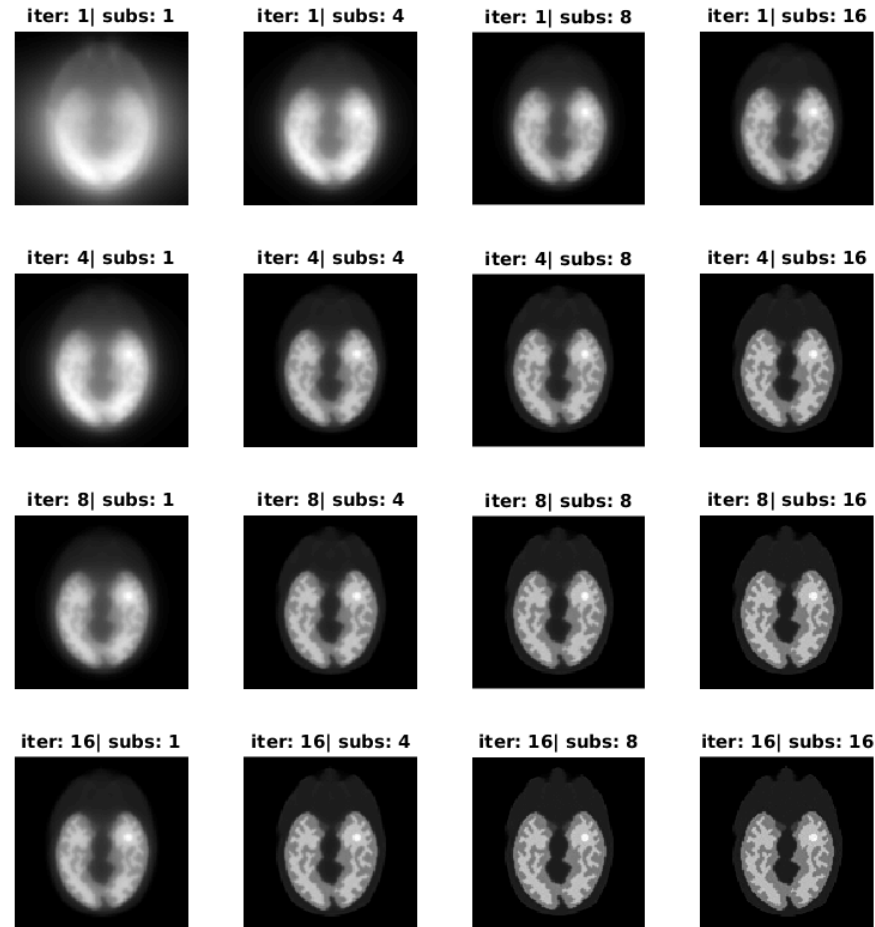
OSEM
rec iter 2/2
subset 25/25



Esercitazione (4)

4. Analisi della relazione tra numero di subset e numero di iterazioni OSEM

Visualizzare il risultato della ricostruzione OSEM usando diverse combinazioni di valori per il numero di subset (fattore di accelerazione) e il numero di iterazioni di ricostruzione.



Esercitazione (4)

5. Valutazione dell'accelerazione ottenuta grazie all'algoritmo OS-EM

Misurare i tempi di esecuzione di una ricostruzione ML-EM (totale, non delle singole sub-iterazioni) e di una ricostruzione OS-EM e verificare che ci sia una velocizzazione del processo di ricostruzione a parità di iterazioni utilizzate.

NOTA BENE:

- mettersi nella condizione $\text{iter_mlem} = \text{iter_osem} * \text{nblock}$
- verificare tutte e 3 le combinazioni possibili
- assicurarsi di misurare l'effettivo tempo di ricostruzione

ART 180 subset 180 iter	5.334660e+00
MLEM 360 iter	3.609439e+00
OSEM 9 subset 40 iter	4.635321e-01
OSEM 40 subset 9 iter	1.287901e-01



Esercitazione (4)

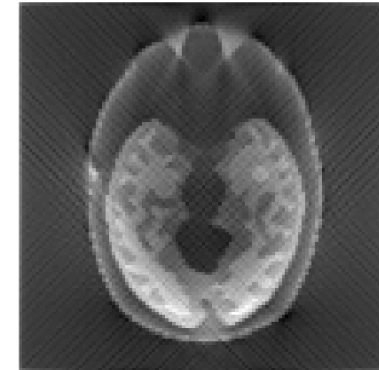
FACOLTATIVO

1. Ricostruzione ART

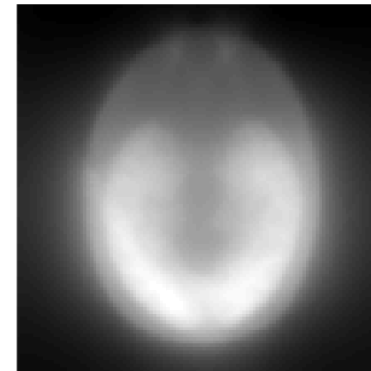
- Utilizzare la funzione creata `Calcolo_Ablock.m` per precalcolare la struttura in blocchi della matrice utilizzata per la ricostruzione ART.
- Seguire il modello dell'implementazione fornita per l'algoritmo OSEM e produrre una implementazione ottimizzata dell'algoritmo ART sviluppato nella precedente esercitazione
- Valutare il tempo necessario alla ricostruzione con questo particolare algoritmo

- ### 2. Ripetere il punto (5) valutando l'andamento dei tempi di calcolo in funzione del numero di iterazioni richieste ai vari algoritmi.

ART
rec iter 180/180



ART
rec iter 1/180



ART
rec iter 180/180

