Introduzione alle tecniche di ricostruzione iterativa per la tomografia ad emissione

Università di Pisa

Focus principale sugli algoritmi ML-EM e OS-EM

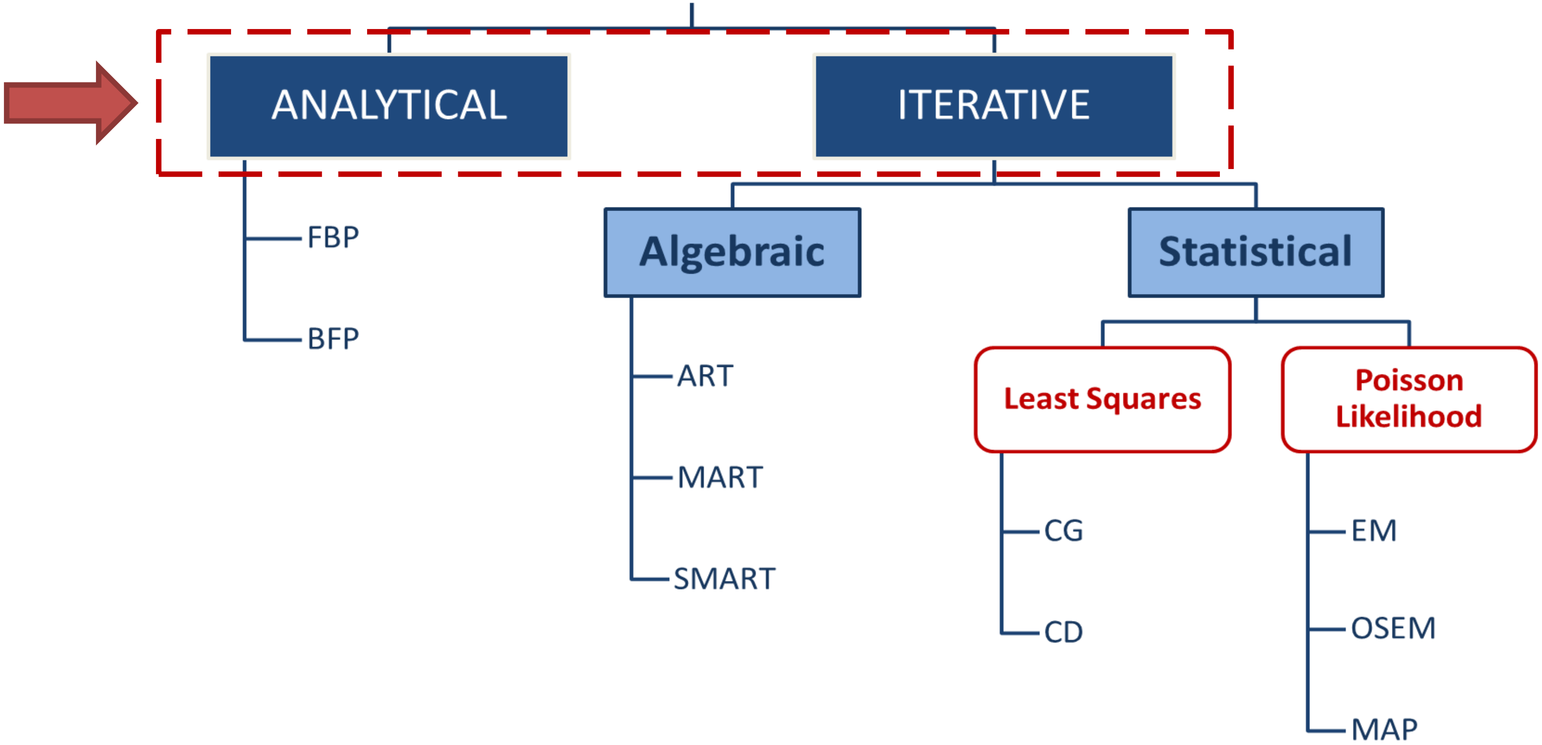
Michele Scipioni
Ph.D. Student
Dip. Ingegneria dell'informazione
Università di Pisa

Corso di Immagini Biomediche, 9 Novembre 2017

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.

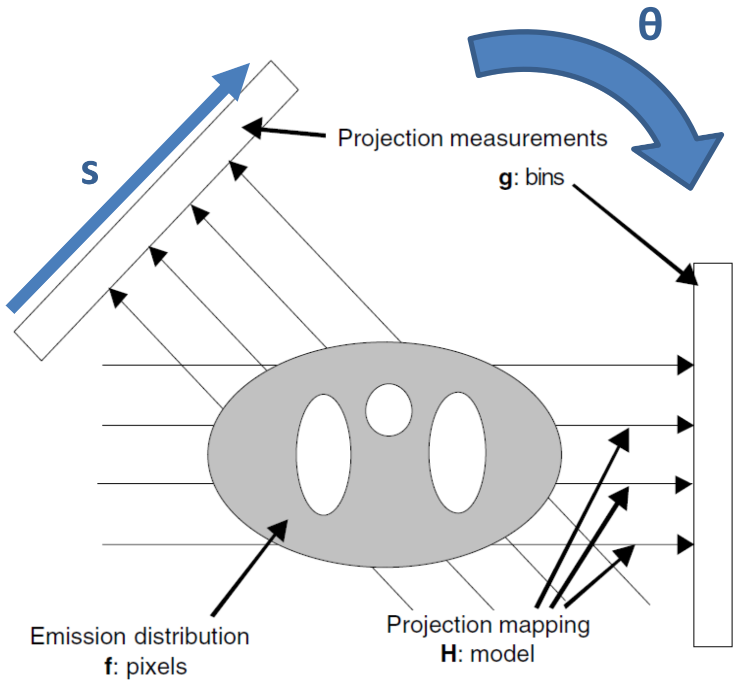
Tecniche analitiche

- 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

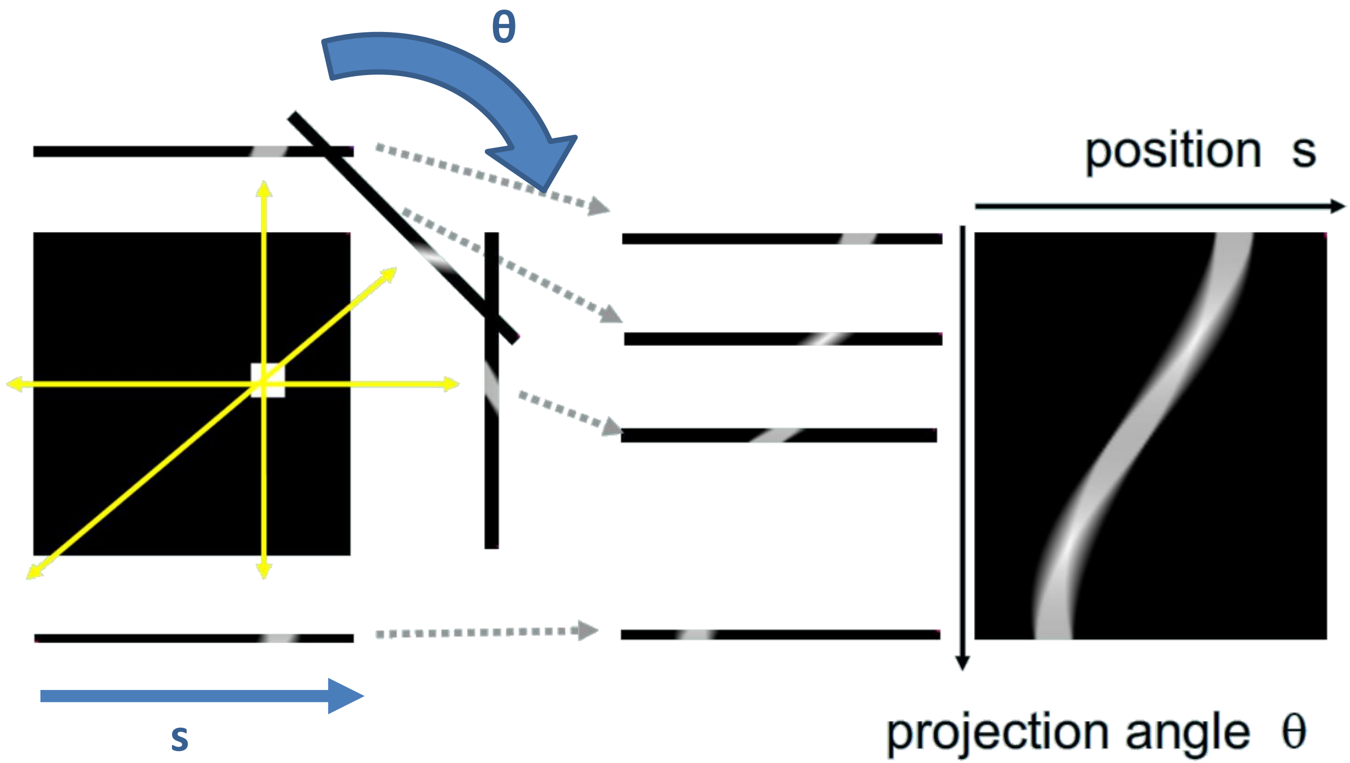


- **g** è l’insieme di proiezioni angolari (θ) misurate, il sinogramma
- **H** è la matrice di sistema, e descrive l'intero processo di registrazione delle immagini. Ogni elemento di H rappresenta il contributo medio di un pixel dell'oggetto al valore di unbin del sinogramma.
- Ogni elemento di **f**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

$g = Hf$ $\quad$ $f = H^{-1}g$

H è grandissima, non quadrata e non invertibile

Modello lineare: il sinogramma



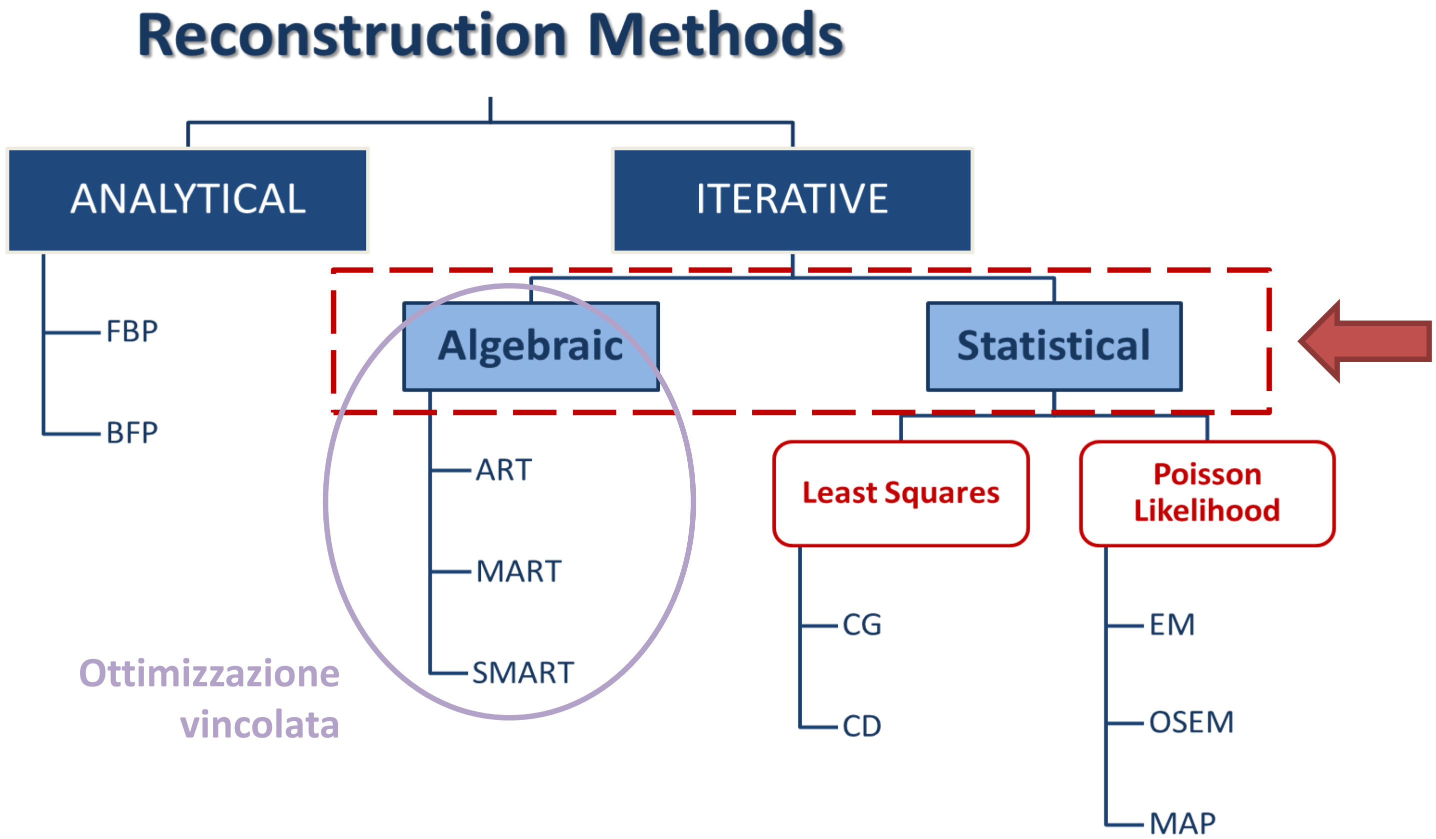
Elementi di un algoritmo iterativo

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

- **criterio di ottimizzazione**: e il criterio rispetto al quale e 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.


Reconstruction Methods
ANALYTICAL
ITERATIVE
FBP
BFP
Algebraic
Statistical
ART
MART
SMART
Least Squares
Poisson Likelihood
CG
CD
EM
OSEM
MAP
Ottimizzazione vincolata

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-negativita 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 e che non offrono nessun meccanismo che consenta di incorporare un modello statistico esplicito dei dati con cui abbiamo a che fare.

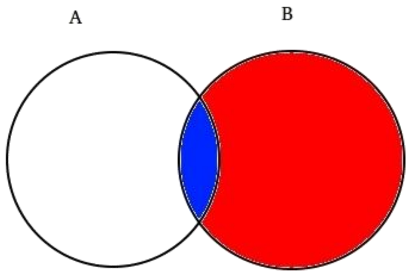
Reconstruction Methods

- ANALYTICAL
  - FBP
  - BFP
- ITERATIVE
  - Algebraic
    - ART
    - MART
    - SMART
  - Statistical
    - Least Squares
      - CG
      - CD
    - Poisson Likelihood
      - EM
      - OSEM
      - MAP

Criterio di massima verosimiglianza

Teorema di Bayes: idea di base

good to know...



PROBABILITA' CONDIZIONATA

La probabilità che si verifichi un evento A, dato B, è uguale alla probabilità che A e B avvengano insieme, diviso la probabilità di B.

$$P(A|B=b) = \frac{BlueArea}{BlueArea + RedArea} = \frac{P(A \cap B)}{P(B)} = \frac{P(B|A)P(A)}{P(B)}$$

Esempio: dato un risultato del lancio (B) qual'è la probabilità che la moneta sia truccata (A)?

- **A** = coeff di affidabilità della moneta (è truccata?)
- **B** = risultato del lancio (testa o croce?)

Teorema di Bayes: interpretazione

good to know…

Esempio: dato un risultato del lancio (B) qual'è la probabilità che la moneta sia truccata (A)?

- **A** = coeff di affidabilità della moneta (è truccata?)
- **B** = risultato del lancio (testa o croce?)

$$\mathbf{P(A|B = b) = \frac{P(B|A)P(A)}{P(B)}}$$

- **P(A) → prior knowledge:** ipotesi a priori circa l'affidabilità della moneta (potremmo scegliere un valore tra 0 ed 1, dove 0.5 indica che la moneta non fa preferenze tra testa e croce
- **P(B|A) → likelihood:** dato un campione di lanci misurati (B), è la probabilità di ottenere (es) quel numero di «teste», data l'ipotesi fatta su A
- **P(B) =** $\sum_i P(B \cap A_i)$ **→ evidence**: probabilità di ottenere il dato misurato dati tutti i possibili valori che può assumere A
- **P(A|B) → posterior belief:** come rivediamo le nostre convinzioni circa la moneta (***P(A)***), dopo aver effettuato una serie di misure

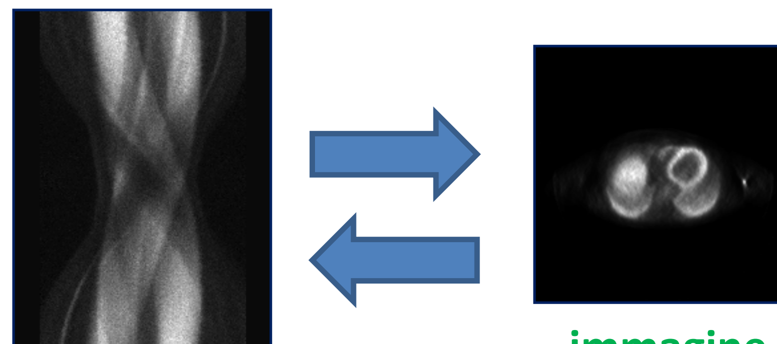
Teorema di Bayes: come lo usiamo?

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad \Rightarrow \quad p(\text{immagine}|\text{dati}) = \frac{p(\text{dati}|\text{immagine})\ p(\text{immagine})}{p(\text{dati})}$$

OBIETTIVO

Vogliamo trovare una stima dell'**immagine** che massimizzi la probabilità a posteriori:

p(immagine|dati)



dati

immagine

Maximum likelihood

- Stima della distribuzione a posteriori → **PROBLEMA INVERSO COMPLESSO**
  p(immagine|dati)
- Stima della likelihood dei dati → **PROBLEMA DIRETTO (PROIEZIONE) «FACILE»**
  p(dati|immagine)

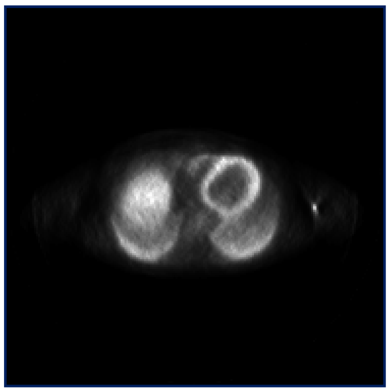
Ipotizziamo per ora che tutte le possibili immagini che possono aver generato i dati sono equiprobabili

Bayes → $p(\text{immagine}|\text{dati}) \sim \dfrac{p(\text{dati}|\text{immagine})\ \cancel{p(\text{immagine})}}{\cancel{p(\text{dati})}}$

i nostri dati sono costanti

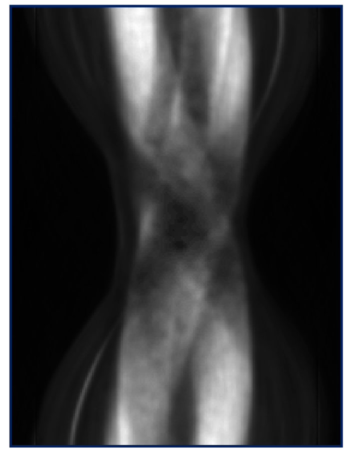
Maximum likelihood

p(immagine|dati) ~ p(dati|immagine)



$\widehat{f}_j$

proiezione



$$\widehat{g}_i = \sum_j h_{ij} \widehat{f}_j + s_i$$

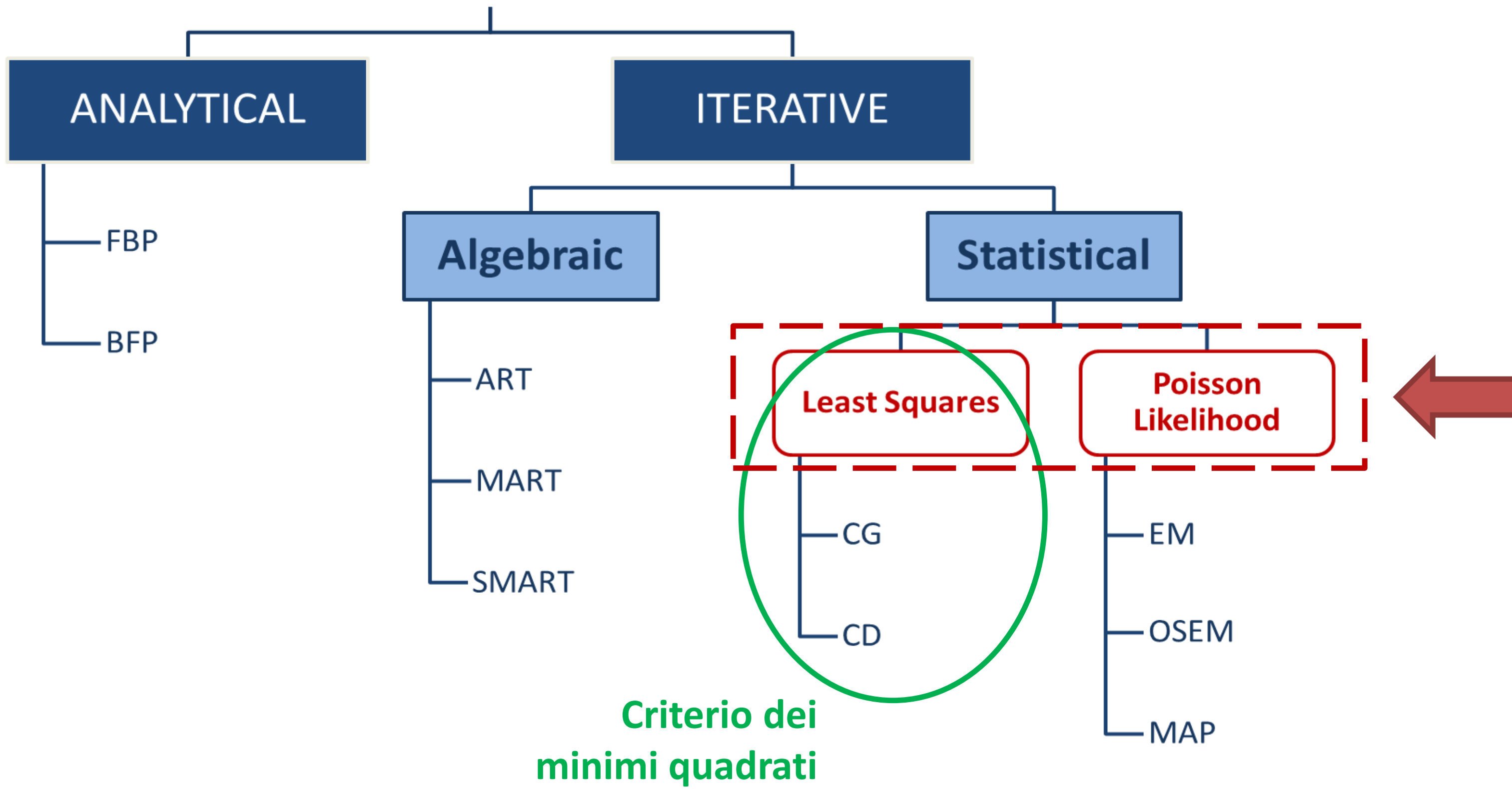
rumore

p(dati|immagine)

$$= \prod_i p(g_i \mid \widehat{g}_i)$$

MODELLO DI RUMORE: gaussiano, poisson, …

Reconstruction Methods



Criterio dei minimi quadrati: likelihood gaussiana

Criterio LS

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

maximize **p (g|Hf)** $= \prod_i \exp\left(-\frac{([Hf]_i - g_i)^2}{2\sigma^2}\right)$ with $[Hf]_i = \sum_j h_{ij} f_j$

or maximize **ln p (g|Hf)** $= \sum_i -\frac{([Hf]_i - g_i)^2}{2\sigma^2}$

or minimize $l = \sum_i ([Hf]_i - g_i)^2 = (Hf - g)'(Hf - g)$

$$\frac{\partial l}{\partial f_j} = 0$$

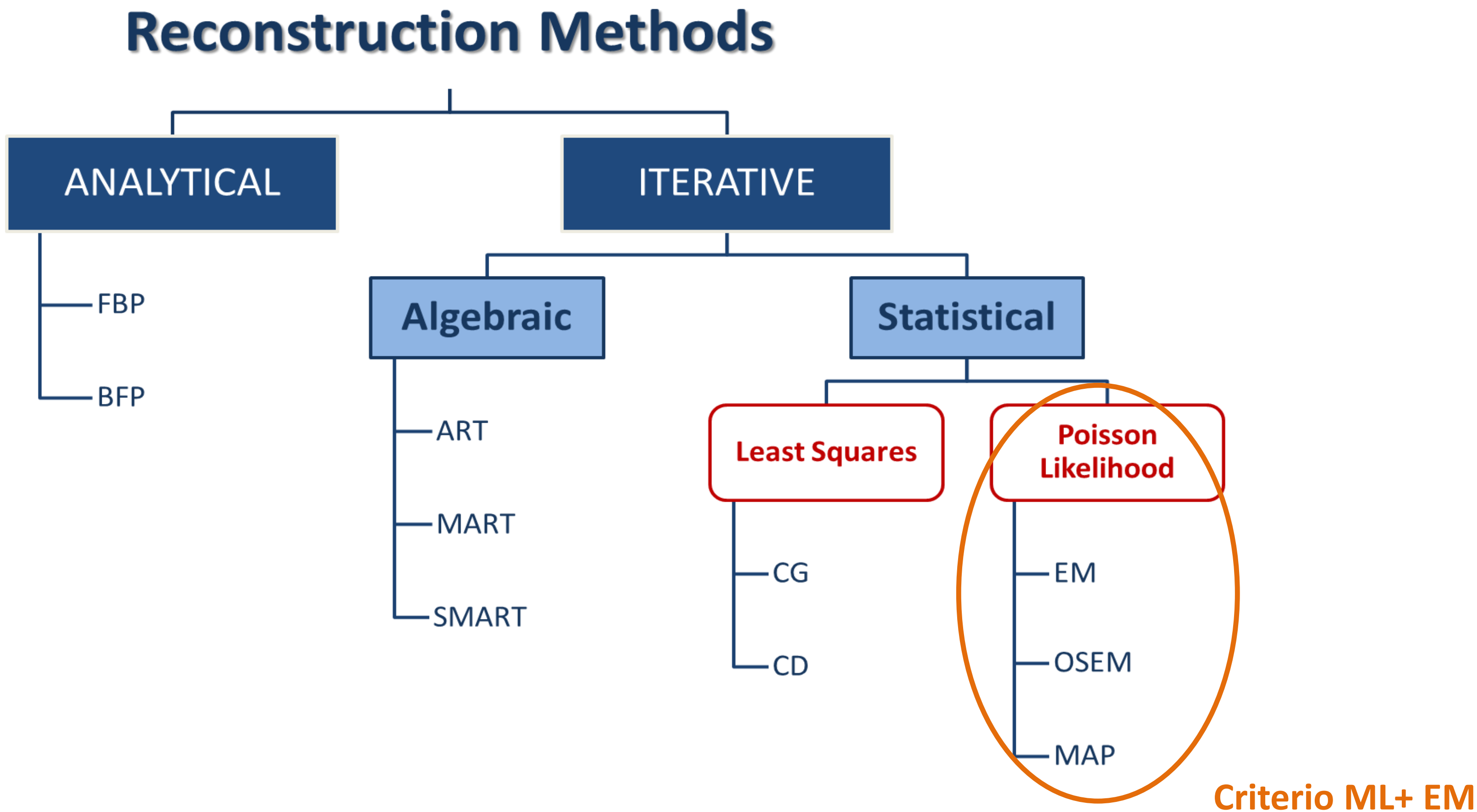
$$\widehat{f} = [H'H]^{-1} H'g$$

Criterio dei minimi quadrati: likelihood gaussiana

H’H è un operatore lineare che calcola la retroproiezione della proiezione

H’H è una matrice gigante che però non modella altro che l’effetto di blurring del sistema.
[H’H]$^{-1}$ di conseguenza cerca di fare deblurring in un modo equivalente al **filtro a rampa**.

$$f = \left[H'H\right]^{-1} H'g$$ **è una backprojection filtered (BPF)**


Reconstruction Methods
ANALYTICAL
ITERATIVE
FBP
BFP
Algebraic
Statistical
ART
MART
SMART
Least Squares
Poisson Likelihood
CG
CD
EM
OSEM
MAP
Criterio ML+ EM

Maximum Likelihood (ML): *dati poissoniani*

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:

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

Valore atteso (medio)

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

Probabilità di misurare g, data un'immagine vera f

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

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

→ ***Aspettazione della log-likelihood del set completo***

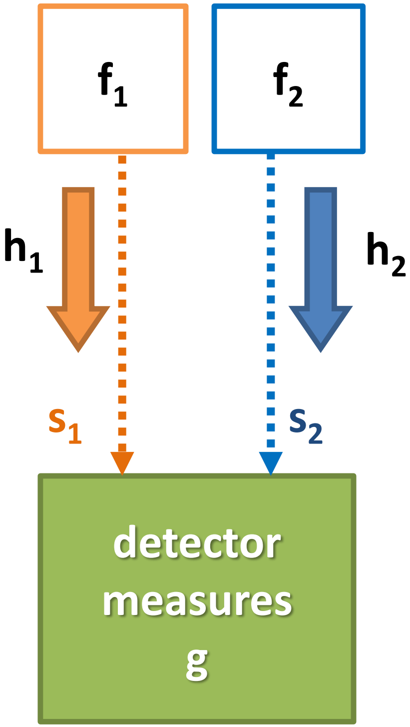
M-step

$$\hat{\boldsymbol{f}}^{(n+1)} = argmax_f\, Q\left(\boldsymbol{f}\middle|\hat{\boldsymbol{f}}^{(n)}\right)$$

→ ***Massimizzazione della stessa***

Derivazione ML-EM: *dataset completo*

good to know...



- due fiale con attività $\mathbf{f_1}$ e $\mathbf{f_2}$
- detettore con sensibilità $\mathbf{h_1}$ ed $\mathbf{h_2}$ rispetto alle due sorgenti

CASO REALE: il sensore misure $g = s_1 + s_2$

DOMANDA: quanti fotoni ha inviato ogni sorgente?

1. $E(s_1) = h_1 f_1$ e $E(s_2) = h_2 f_2$
2. $g = s_1 + s_2$

SOLUZIONE:

$$E\left(s_i \mid g\right) = g \, \frac{h_1 f_i}{h_1 f_1 + h_2 f_2}$$

Derivazione ML-EM: *E-step*

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

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

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

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

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

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

E-step

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

Derivazione ML-EM: *M-step*

E-step

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

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

M-step

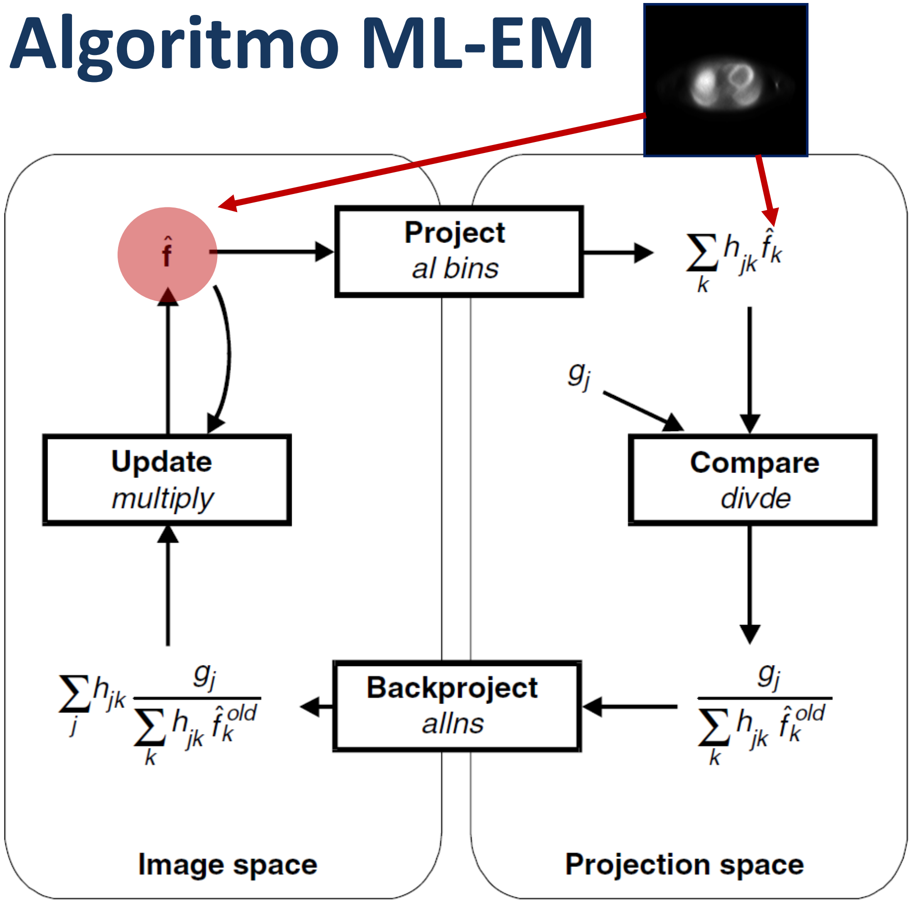
$$\frac{\partial Q\left(\boldsymbol{f}\middle|\hat{\boldsymbol{f}}^{(n)}\right)}{\partial f_j} = 0 = \sum_i \frac{p_{ij}}{\hat{f}_j^{(n+1)}} - \sum_i h_{ij}$$

$$\hat{f}_j^{(n+1)} \sum_i h_{ij} = \sum_i p_{ij} \quad \xrightarrow{\text{yields}}$$

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

M-step

Algoritmo ML-EM



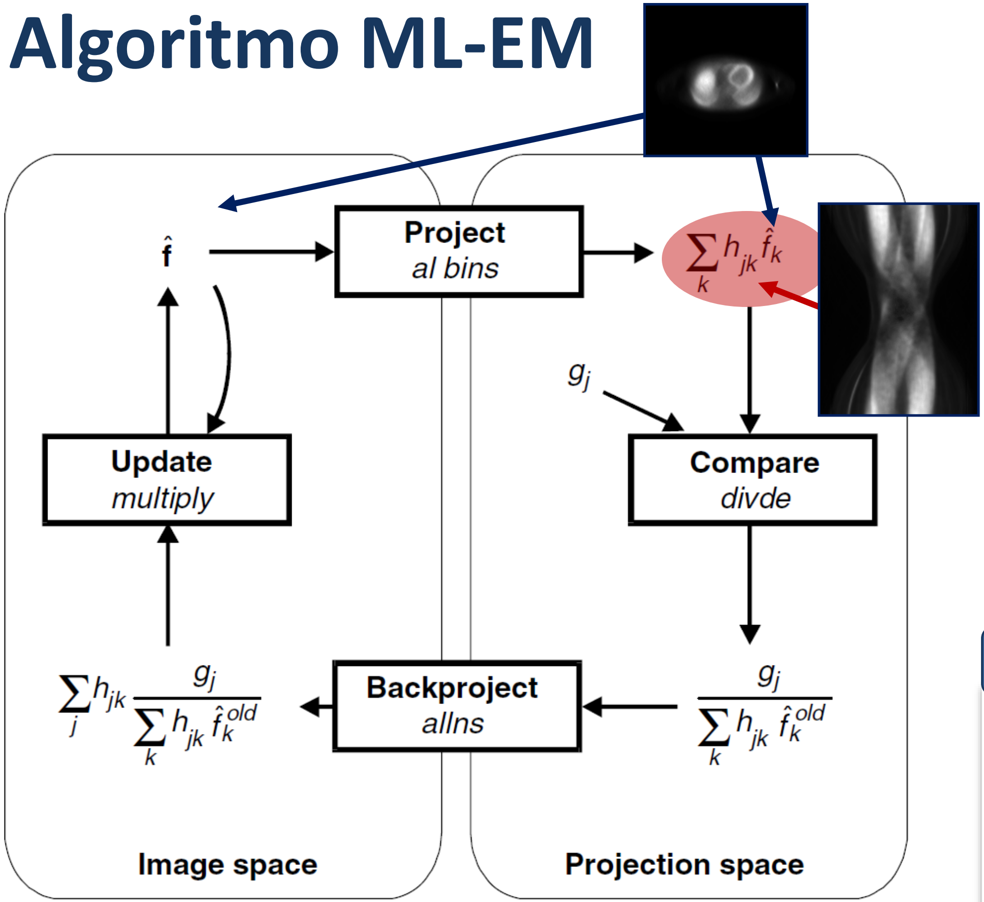
1) **Stima iniziale uniforme dell'immagine**

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

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



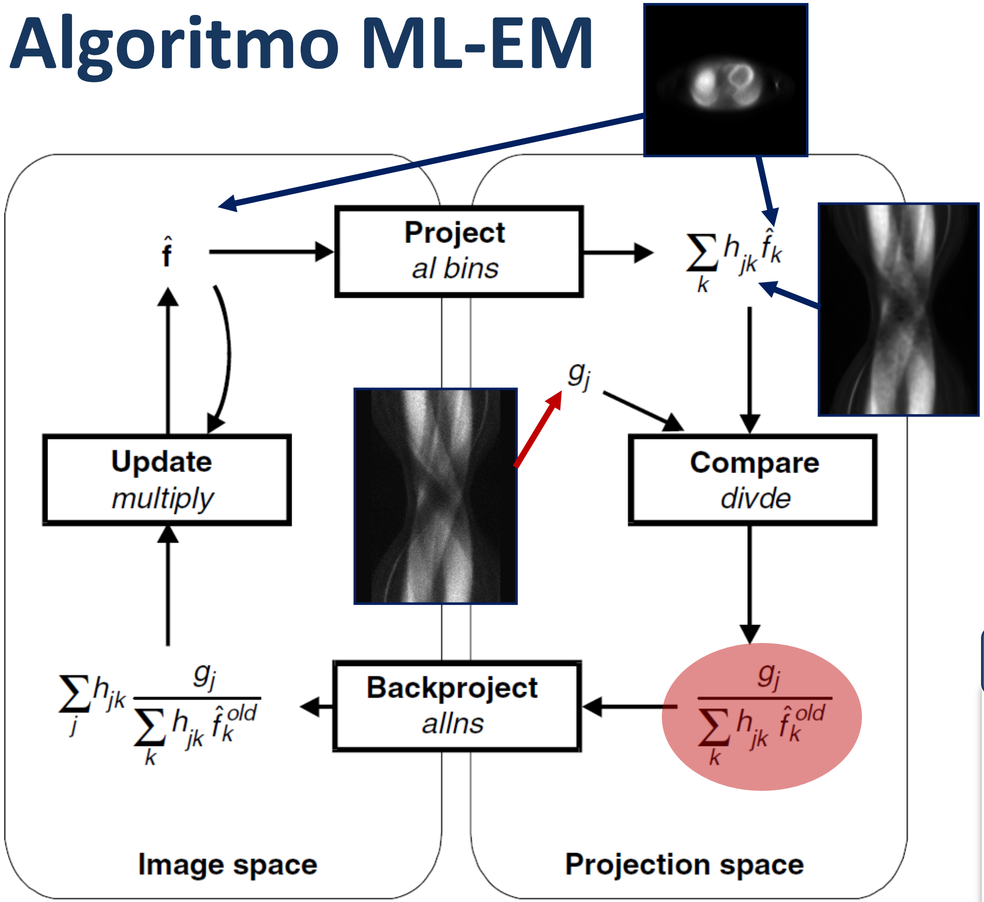
2) **Simulazione della misura a partire dalla stima (*proiezione*)**

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

ML-EM

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

Algoritmo ML-EM



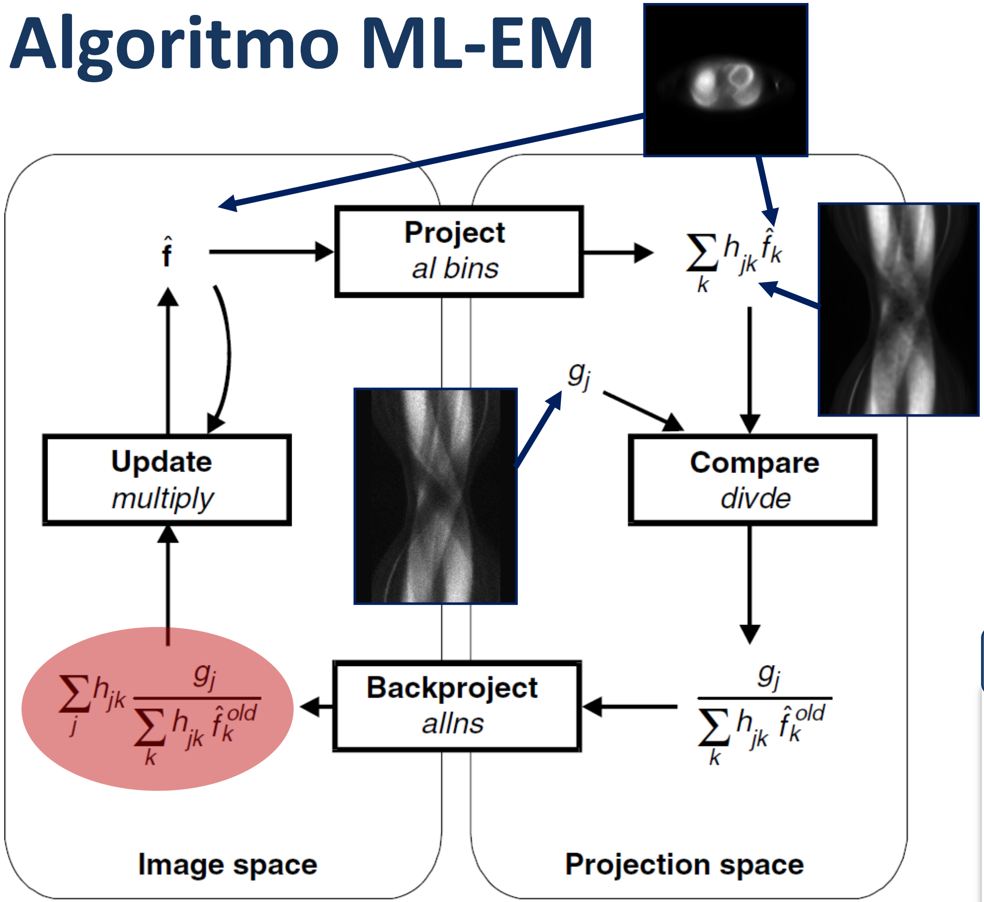
3) **Confronto con la misura vera**

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

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



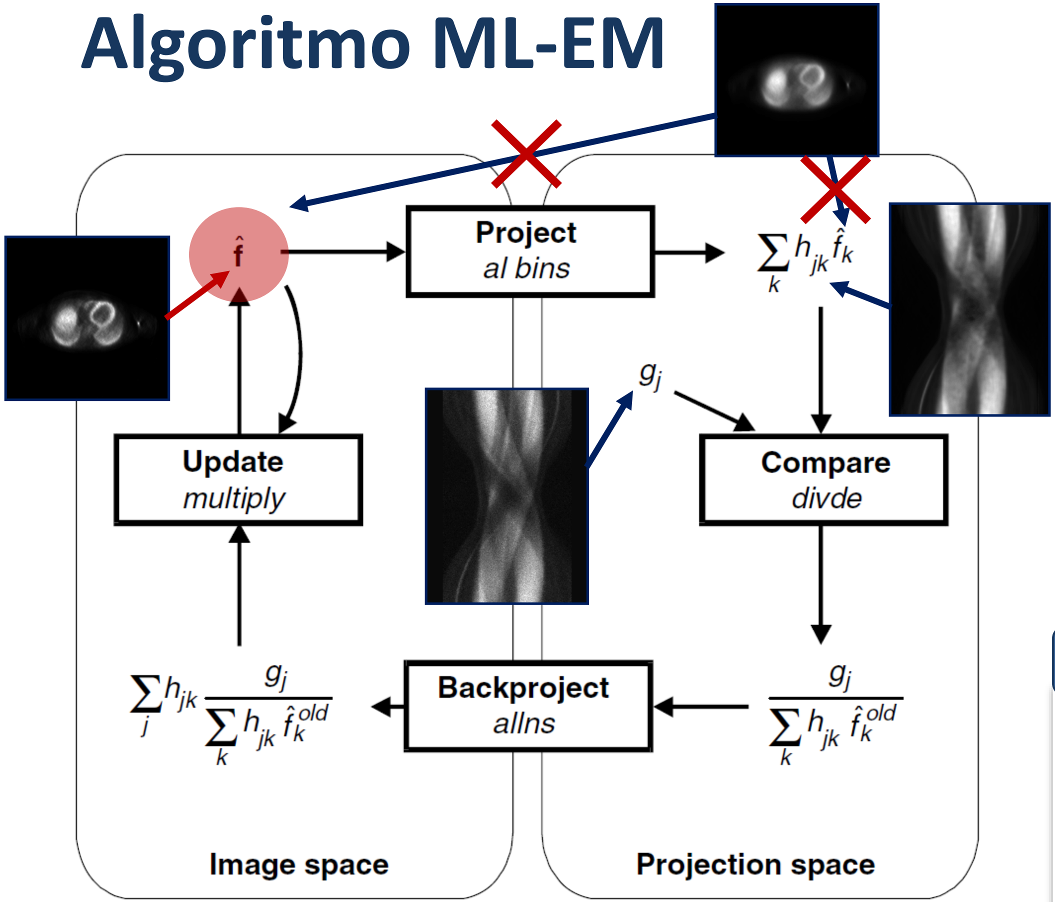
4) **Passaggio al dominio immagine (*backprojection*)**

$$e_i^{(f)} = \sum_i h_{ij}\, e_i^{(g)}$$

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



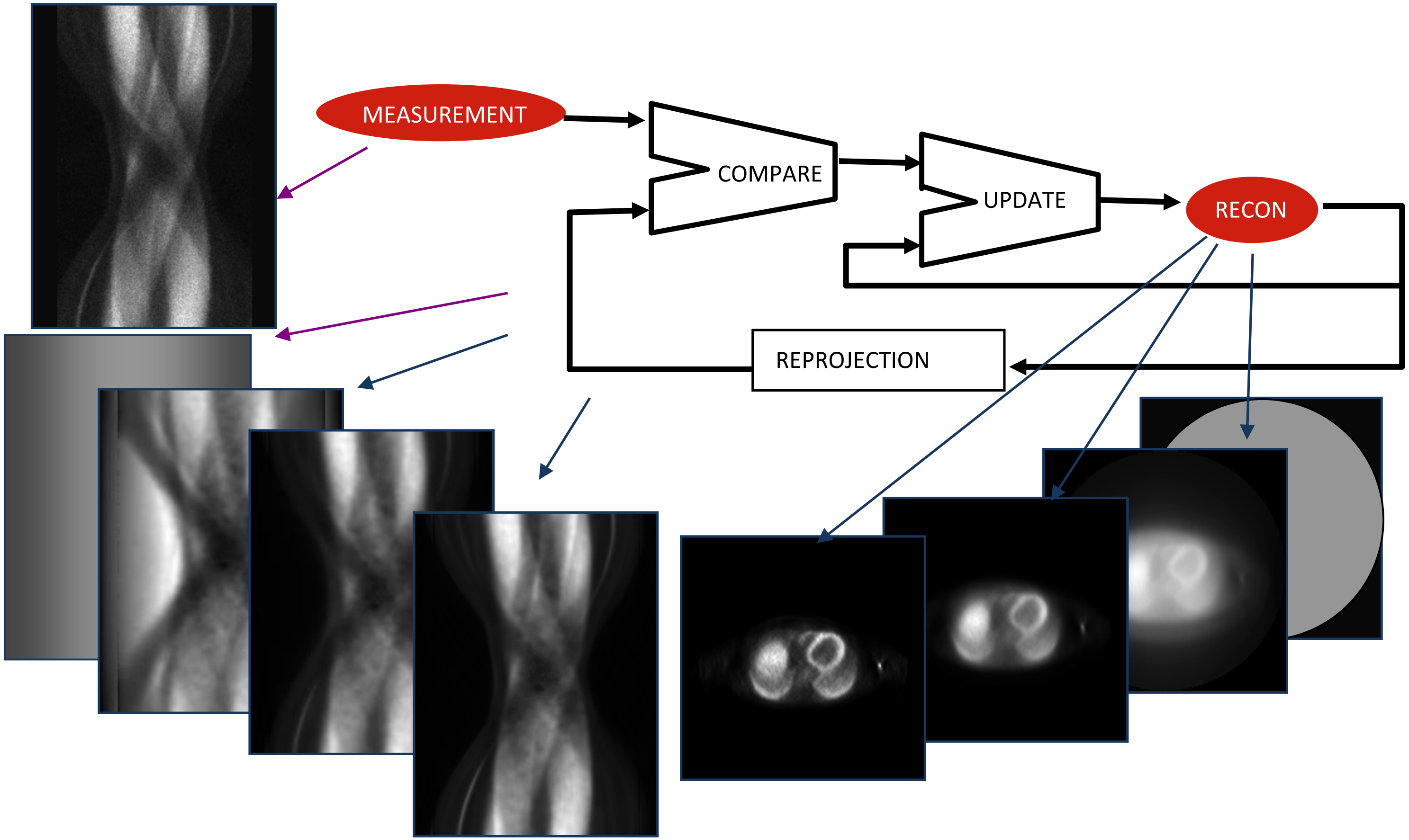
5) Aggiornamento della stima
(*pesatura della stima precedente*)

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\underbrace{\sum_i h_{ij}}_{\sum_i h_{ij} \equiv H'\mathbf{1} \rightarrow sensitivity}} e_i(f)$$

ML-EM

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

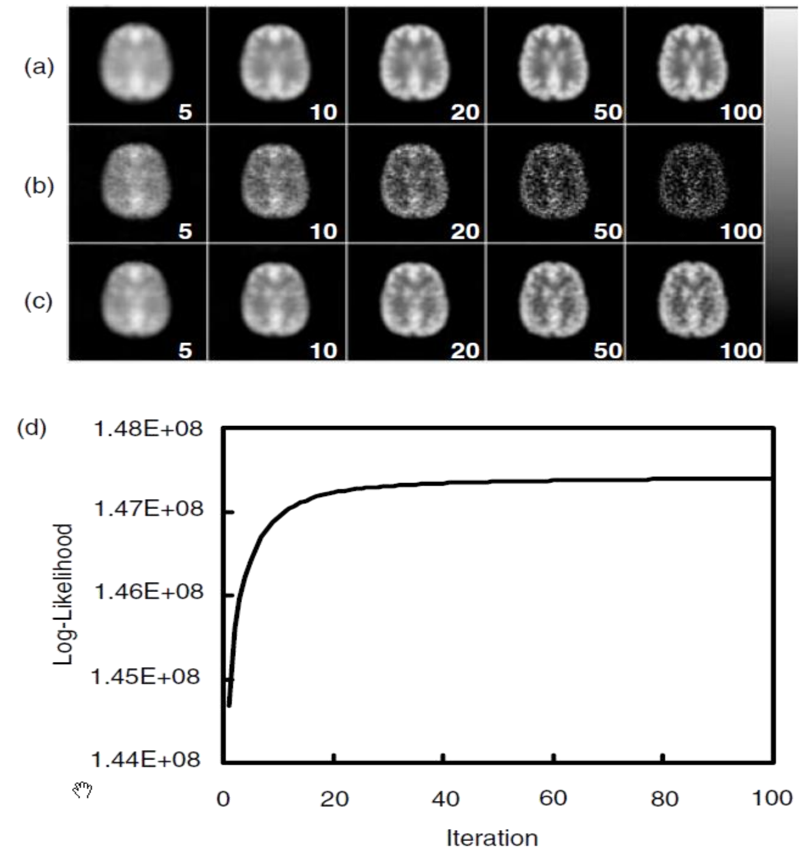
Algoritmo ML-EM



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 a 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*

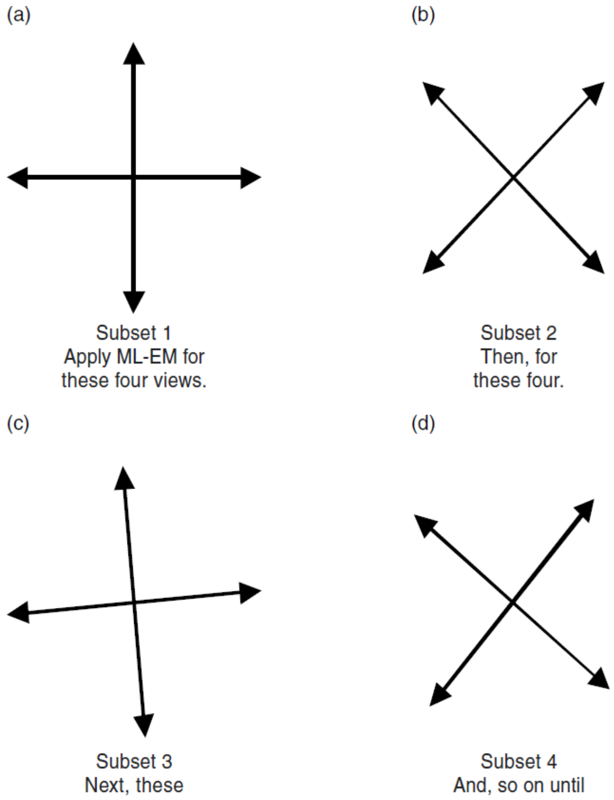
$$\underbrace{\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}}_{\textbf{ML-EM}} \Rightarrow \underbrace{\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i' \epsilon \boldsymbol{S_{n+1}}} h_{i'j}} \sum_{i \epsilon \boldsymbol{S_{n+1}}}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

- La retroproiezione viene applicata soltanto ai bin che appartengono al sotto insieme (***subset***) $\boldsymbol{S_n}$ 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):

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

Algoritmo OS-EM: *organizzazione dei subsets*

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

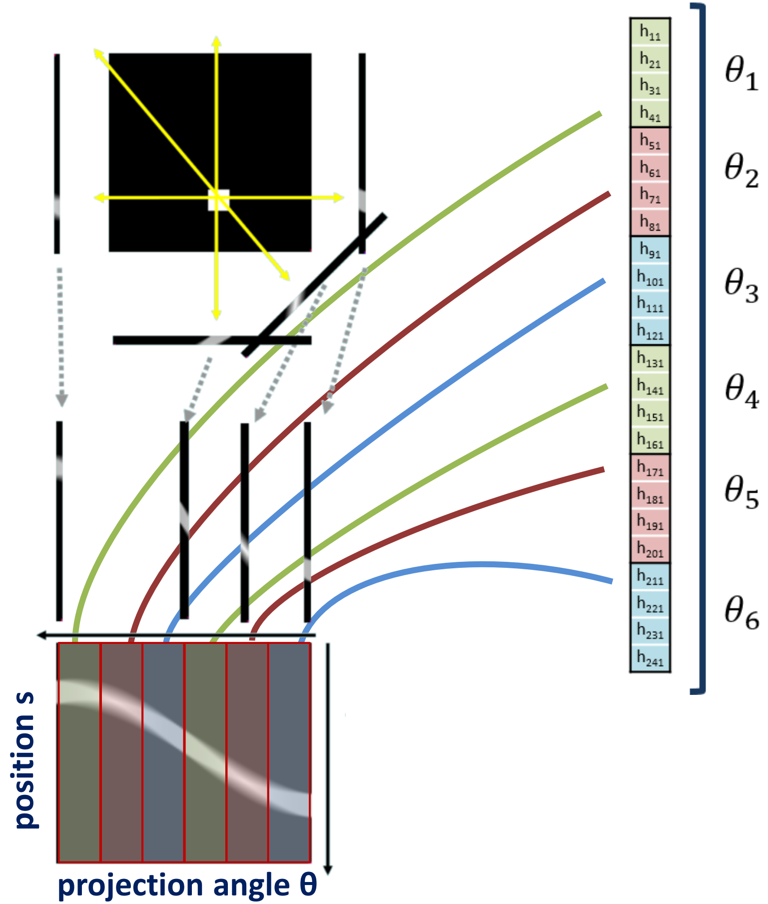


PRINCIPALI REGOLE DI SCELTA

- 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 successive**
- In ogni subset deve comparire il contributo di **tutti i pixel del FOV**

Algoritmo OS-EM: *organizzazione dei subsets*

good to know...



CONVERTIRE UNA COPPIA DI COORDINATE 2D IN UN INDICE LINEARE:

Matlab:

idx = sub2ind([#rows, #cols], id_row, id_col);

Manual:

idx = id_row + (id_col-1)#rows;*

SELEZIONARE UNA COLONNA DEL SINOGRAMMA, FISSATO θ (= id_col):

Matlab:

idx_start = sub2ind ([#rows, #cols], 1, id_col);

idx_stop = sub2ind ([#rows, #cols], #rows, id_col);

Manual:

idx_start = 1 + (id_col-1)#rows;*

idx_stop = #rows + (id_col-1)#rows = id_col*#rows;*

Algoritmo OS-EM: *organizzazione dei subsets*

$$f \xrightarrow[3x3]{} \begin{bmatrix} f_1 & f_4 & f_7 \\ f_2 & f_5 & f_8 \\ f_3 & f_6 & f_9 \end{bmatrix}$$

$$H_{24x9}$$

$$\begin{array}{c} \\ \theta_1 \\ \\ \\ \\ \theta_2 \\ \\ \\ \\ \theta_3 \\ \\ \\ \\ \theta_4 \\ \\ \\ \\ \theta_5 \\ \\ \\ \\ \theta_6 \\ \\ \\ \end{array}
\begin{bmatrix}
h_{11} & h_{12} & h_{13} & h_{14} & h_{15} & h_{16} & h_{17} & h_{18} & h_{19} \\
h_{21} & h_{22} & h_{23} & h_{24} & h_{25} & h_{26} & h_{27} & h_{28} & h_{29} \\
h_{31} & h_{32} & h_{33} & h_{34} & h_{35} & h_{36} & h_{37} & h_{38} & h_{39} \\
h_{41} & h_{42} & h_{43} & h_{44} & h_{45} & h_{46} & h_{47} & h_{48} & h_{49} \\
h_{51} & h_{52} & h_{53} & h_{54} & h_{55} & h_{56} & h_{57} & h_{58} & h_{59} \\
h_{61} & h_{62} & h_{63} & h_{64} & h_{65} & h_{66} & h_{67} & h_{68} & h_{69} \\
h_{71} & h_{72} & h_{73} & h_{74} & h_{75} & h_{76} & h_{77} & h_{78} & h_{79} \\
h_{81} & h_{82} & h_{83} & h_{84} & h_{85} & h_{86} & h_{87} & h_{88} & h_{89} \\
h_{91} & h_{92} & h_{93} & h_{94} & h_{95} & h_{96} & h_{97} & h_{98} & h_{99} \\
h_{101} & h_{102} & h_{103} & h_{104} & h_{105} & h_{106} & h_{107} & h_{108} & h_{109} \\
h_{111} & h_{112} & h_{113} & h_{114} & h_{115} & h_{116} & h_{117} & h_{118} & h_{119} \\
h_{121} & h_{122} & h_{123} & h_{124} & h_{125} & h_{126} & h_{127} & h_{128} & h_{129} \\
h_{131} & h_{132} & h_{133} & h_{134} & h_{135} & h_{136} & h_{137} & h_{138} & h_{139} \\
h_{141} & h_{142} & h_{143} & h_{144} & h_{145} & h_{146} & h_{147} & h_{148} & h_{149} \\
h_{151} & h_{152} & h_{153} & h_{154} & h_{155} & h_{156} & h_{157} & h_{158} & h_{159} \\
h_{161} & h_{162} & h_{163} & h_{164} & h_{165} & h_{166} & h_{167} & h_{168} & h_{169} \\
h_{171} & h_{172} & h_{173} & h_{174} & h_{175} & h_{176} & h_{177} & h_{178} & h_{179} \\
h_{181} & h_{182} & h_{183} & h_{184} & h_{185} & h_{186} & h_{187} & h_{188} & h_{189} \\
h_{191} & h_{192} & h_{193} & h_{194} & h_{195} & h_{196} & h_{197} & h_{198} & h_{199} \\
h_{201} & h_{202} & h_{203} & h_{204} & h_{205} & h_{206} & h_{207} & h_{208} & h_{209} \\
h_{211} & h_{212} & h_{213} & h_{214} & h_{215} & h_{216} & h_{217} & h_{218} & h_{219} \\
h_{221} & h_{222} & h_{223} & h_{224} & h_{225} & h_{226} & h_{227} & h_{228} & h_{229} \\
h_{231} & h_{232} & h_{233} & h_{234} & h_{235} & h_{236} & h_{237} & h_{238} & h_{239} \\
h_{241} & h_{242} & h_{243} & h_{244} & h_{245} & h_{246} & h_{247} & h_{248} & h_{249}
\end{bmatrix}$$

$$H_1 f = g_1$$

$$H_2 f = g_2$$

$$H_3 f = g_3$$

$\mathrm{N}_{\mathrm{proiezioni}}$: 6

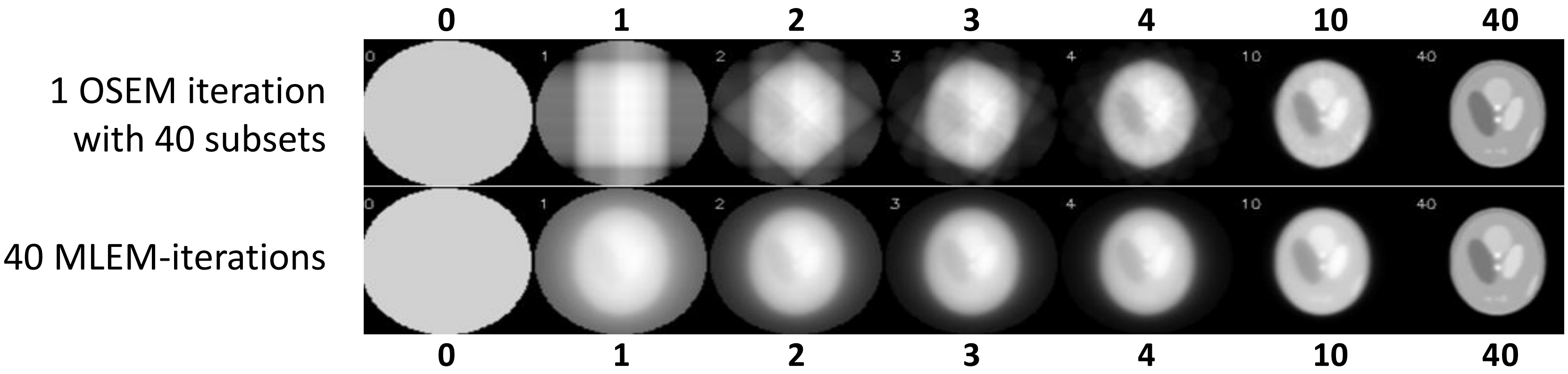
$\mathrm{N}_{\mathrm{posizioni}}$: 4

$g \rightarrow [4x6]$

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

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 $(N_{subset} \text{ x } N)$ 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)

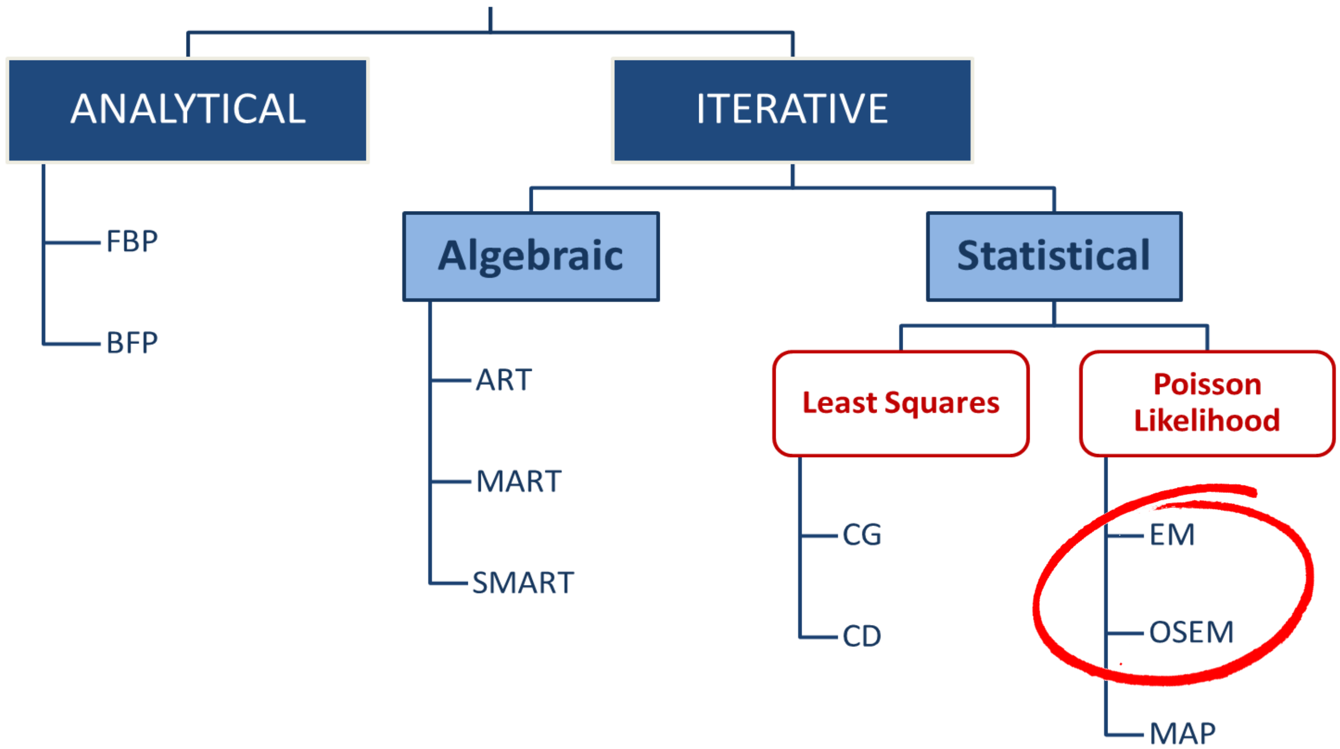
Abbiamo detto precedentemente che se il sistema di imaging corregge internamente i sinogrammi prima che avvenga la ricostruzione, la statistica muta e non sarebbe più valido lavorare nell'ipotesi di Poissonianità dei dati su cui si basano ML-EM ed OS-EM.

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 corregge le immagini allo stesso tempo.

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

$$\textbf{OS-EM} \left\{ \hat{f_k}^{(n+1)} = \frac{\hat{f_k}^{(n)}}{{H_k}^T 1_k} {H_k}^T \frac{g_k}{att_k.* \left(H_k\hat{f_k}^{(n)}\right) + scatt_k + rand_k} \right.$$

Reconstruction Methods



Limite intrinseco degli approcci ML, LS e WLS

Sfortunamente 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)

$$\hat{\mathrm{f}} = \mathrm{H}^{+}\mathrm{g} = \mathrm{H}^{+}(\mathrm{Hf} + \mathrm{n}) = \mathrm{f} + \mathrm{H}^{+}\mathrm{n}$$

- 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.

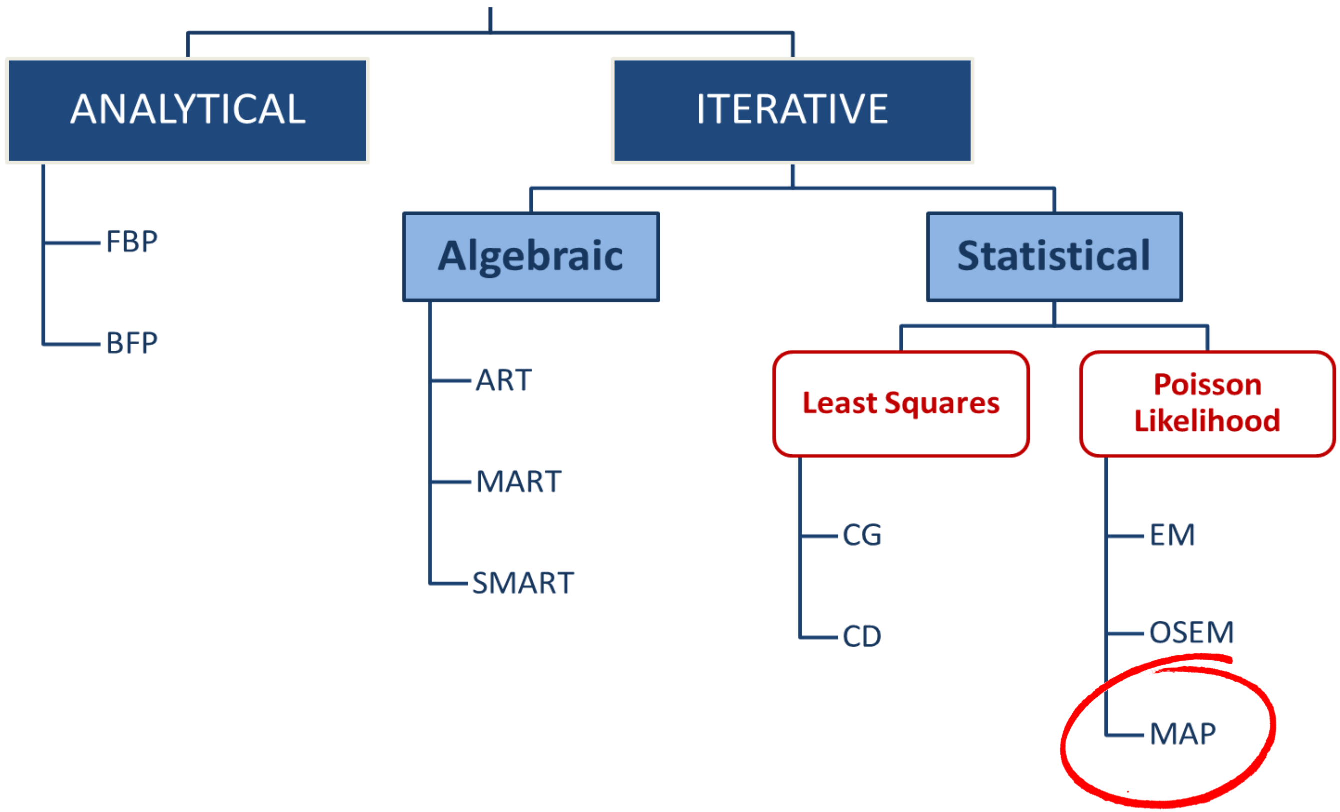
Lo *smoothing* può essere introdotto in due modi:

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

Approcci Bayesiani di stima regolarizzata (Maximum A Posteriori)

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

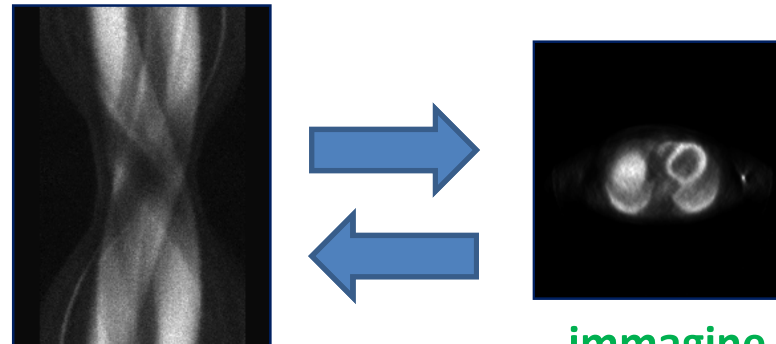
Reconstruction Methods



Problema inverso: maximum a posteriori (MAP)

OBIETTIVO
Vogliamo trovare una stima dell'**immagine** che massimizzi la probabilità a posteriori:

p(immagine|dati)



Bayes →
$$p(\text{immagine}|\text{dati}) \sim \frac{p(\text{dati}|\text{immagine})\ \cancel{p(\text{immagine})}}{\cancel{p(\text{dati})}}$$

Problema inverso: maximum a posteriori (MAP)

Bayes

p(immagine|dati) ~ p(dati|immagine) p(immagine)

ln p(immagine|dati) ~ ln p(dati|immagine) + ln p(immagine)

(posterior) *(likelihood)* *(prior or penalty)*

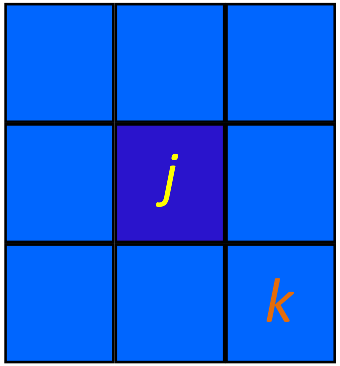
Local prior

$p(f_j \mid f) = p(f_j \mid f_k$, k is neighbor of j)

Gibbs distribution:

$$p(f_j \mid f) = \frac{1}{Z}\exp\left(-\beta\, E_j(N_j)\right)$$

$$\ln p(f_j \mid f) = -\beta\, E_j(N_j) + const = -\beta \sum_{k \in N_j} E(f_j - f_k)$$



Ricostruzione MAP: approssimazione One Step Late

Log-likelihood Poissoniana con prior (log-likelihood penalizzata)

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

Exact MAP

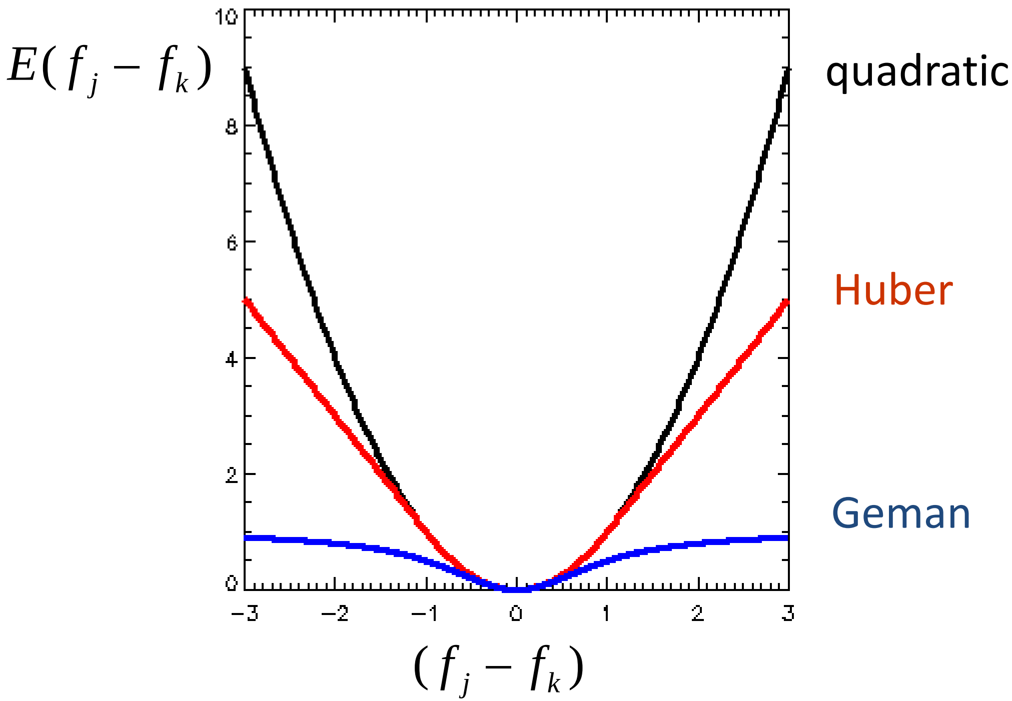
$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j} - \boldsymbol{\beta} \left.\frac{\boldsymbol{\partial E(f)}}{\boldsymbol{\partial f_j}}\right|_{\cancel{\hat{f}_j^{(n+1)}}}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}$$

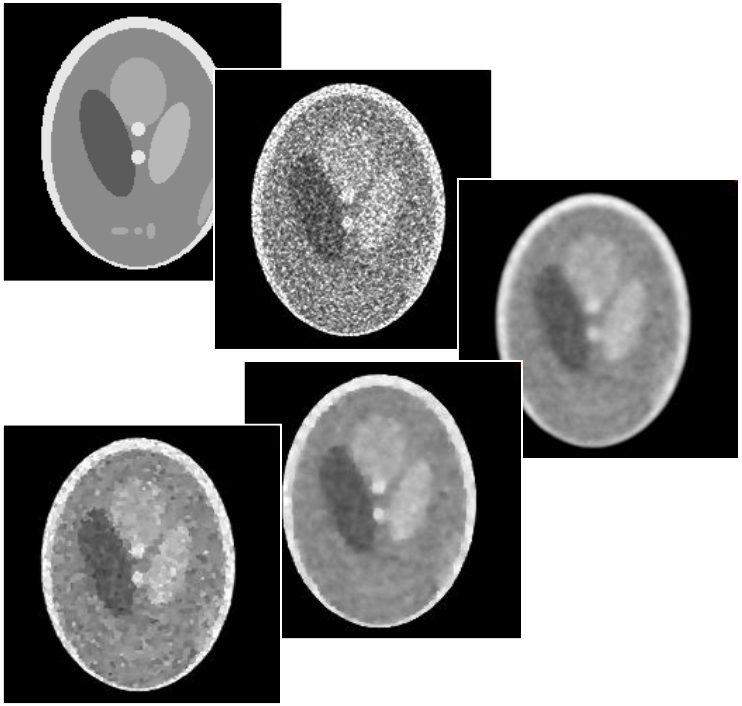
One Step Late (OSL)

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j} - \boldsymbol{\beta} \left.\frac{\boldsymbol{\partial E(f)}}{\boldsymbol{\partial f_j}}\right|_{\hat{f}_j^{(n)}}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}$$

Effetto della scelta del prior: local smoothing

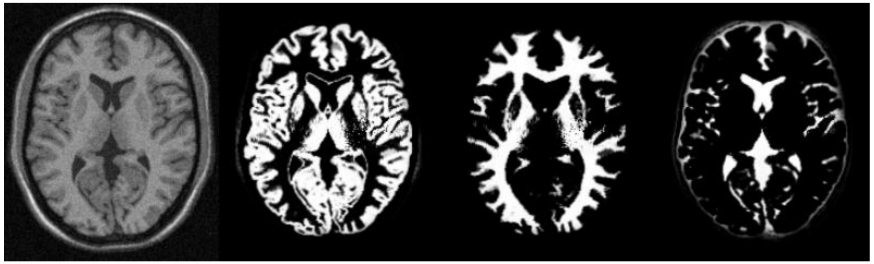
$$\ln p(f_j \mid f) = -\beta\, E_j(N_j) + const = -\beta \sum_{k \in N_j} E(f_j - f_k)$$



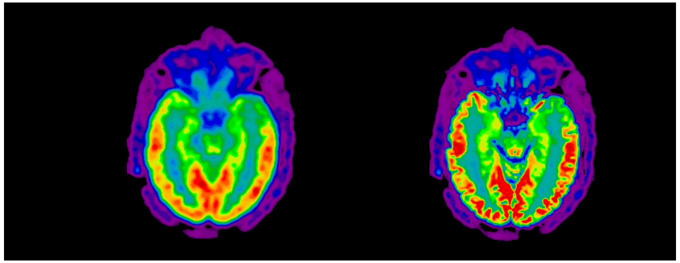


Effetto della scelta del prior: info anatomiche

Segmented MRI: quando si analizza il vicinato, si fa smoothing solo tra voxel dello stesso cluster

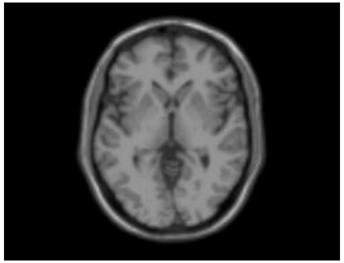


T1 Grey White CSF

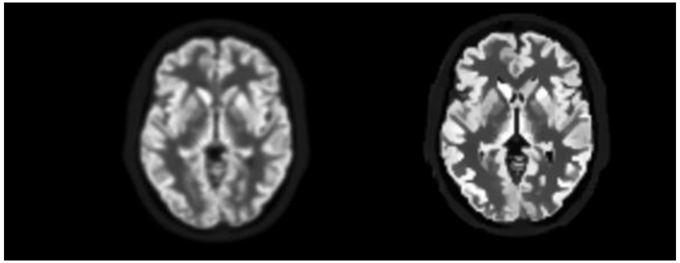


MLEM + smoothing MAP-OSL

No segmentation: si usano direttamente i livelli di grigio della T1 per identificare voxel "simili"



T1



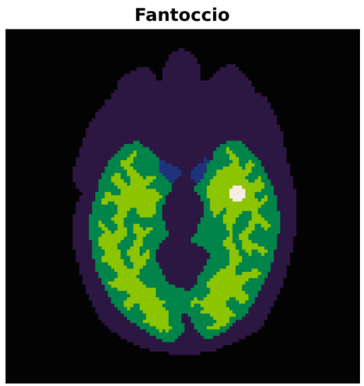
MLEM + smoothing MAP-OSL

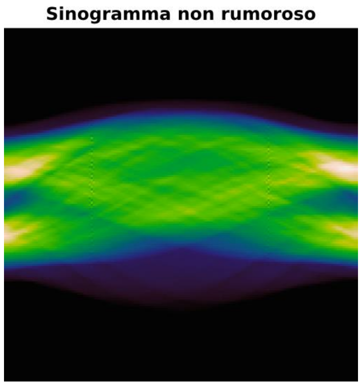
Esercitazione

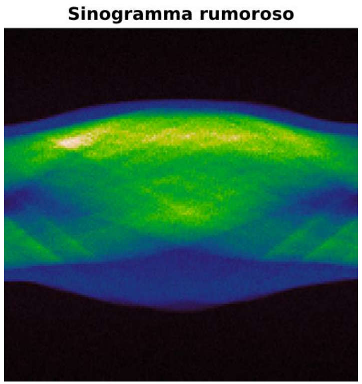
Esercitazione (0)

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_Hblock.m*** con descrizione di quale debba essere la sua struttura e il suo funzionamento.







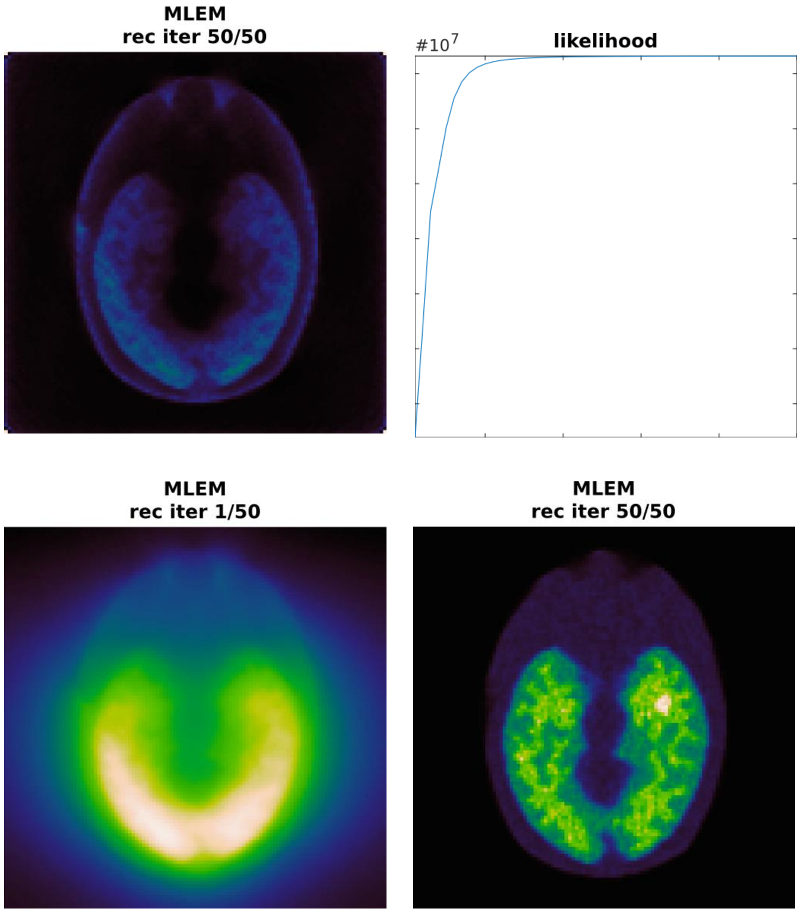
Esercitazione (1)

1. Ricostruzione ML-EM

a) Implementare l'algoritmo ML-EM per la ricostruzione del sinogramma rumoroso generato ai punti precedenti.

b) Valutare la qualità della ricostruzione a seconda che vengano corretti o meno i disturbi simulati.

c) Salvare l'immagine intermedia ricostruita ad ogni iterazione in un vettore 3D (N,N,iter_mlem)



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 corregge le immagini allo stesso tempo.

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

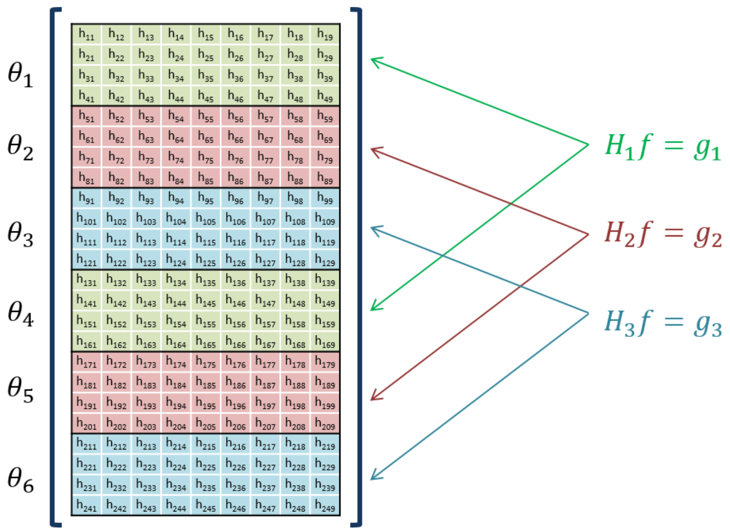
$$\text{OS-EM} \left\{ \hat{f}_k^{(n+1)} = \frac{\hat{f}_k^{(n)}}{{H_k}^T 1_k} {H_k}^T \frac{g_k}{att_k.*\left(H_k\hat{f}_k^{(n)}\right) + scatt_k + rand_k} \right.$$

Esercitazione (2)

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

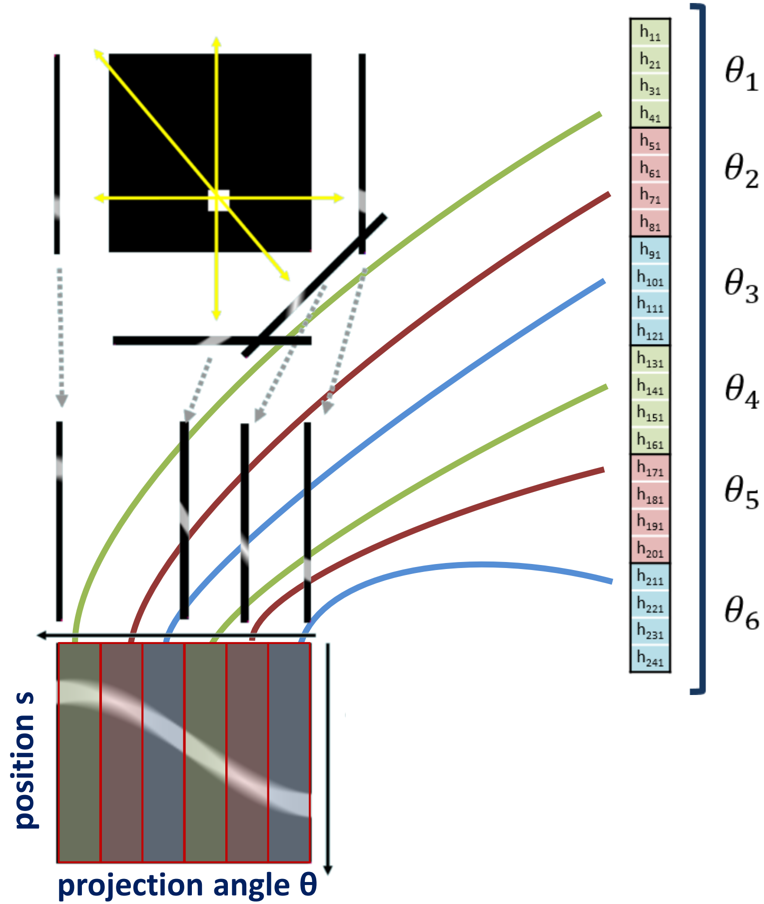
a) Creare una funzione esterna (partire da Calcolo Hblock.m fornito) per l'estrazione dei blocchi della matrice di sistema con cui ricostruire i singoli subset del sinogramma.

b) La descrizione della funzione e fornita nel file 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 e associato un determinato blocco.



Algoritmo OS-EM: *organizzazione dei subsets*

good to know...



CONVERTIRE UNA COPPIA DI COORDINATE 2D IN UN INDICE LINEARE:

Matlab:

idx = sub2ind([#rows, #cols], id_row, id_col);

Manual:

idx = id_row + (id_col-1)#rows;*

SELEZIONARE UNA COLONNA DEL SINOGRAMMA, FISSATO θ (= id_col):

Matlab:

idx_start = sub2ind ([#rows, #cols], 1, id_col);

idx_stop = sub2ind ([#rows, #cols], #rows, id_col);

Manual:

idx_start = 1 + (id_col-1)#rows;*

idx_stop = #rows + (id_col-1)#rows = id_col*#rows;*

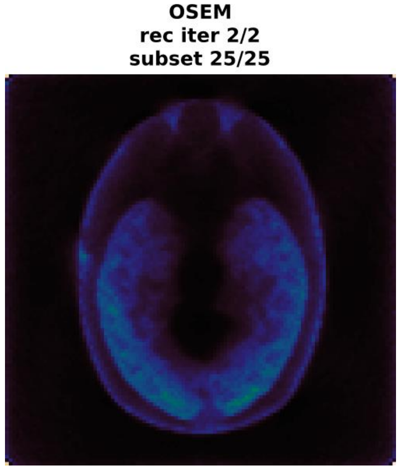
Esercitazione (3)

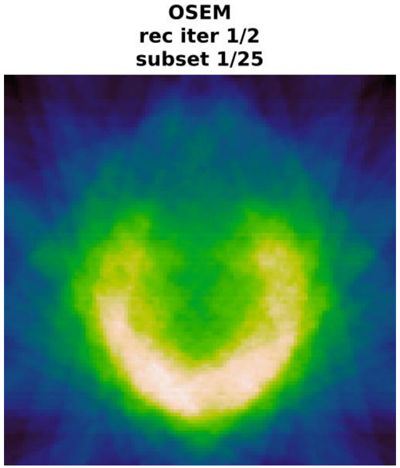
1. Ricostruzione OS-EM

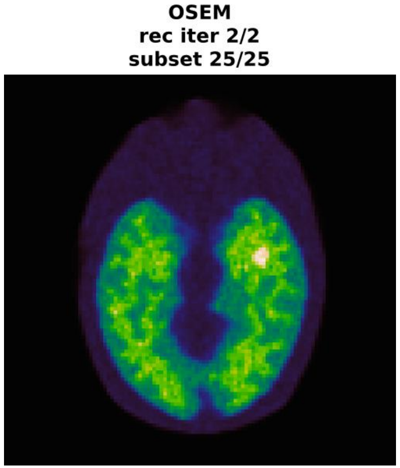
a) Implementare l'algoritmo OS-EM per la ricostruzione del sinogramma rumoroso generato ai punti precedenti.

b) Valutare la qualità della ricostruzione a seconda che vengano corretti o meno i disturbi simulati.

c) Salvare l'immagine intermedia ricostruita ad ogni iterazione in un vettore 3D (N,N,iter_mlem*nblock)







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 corregge le immagini allo stesso tempo.

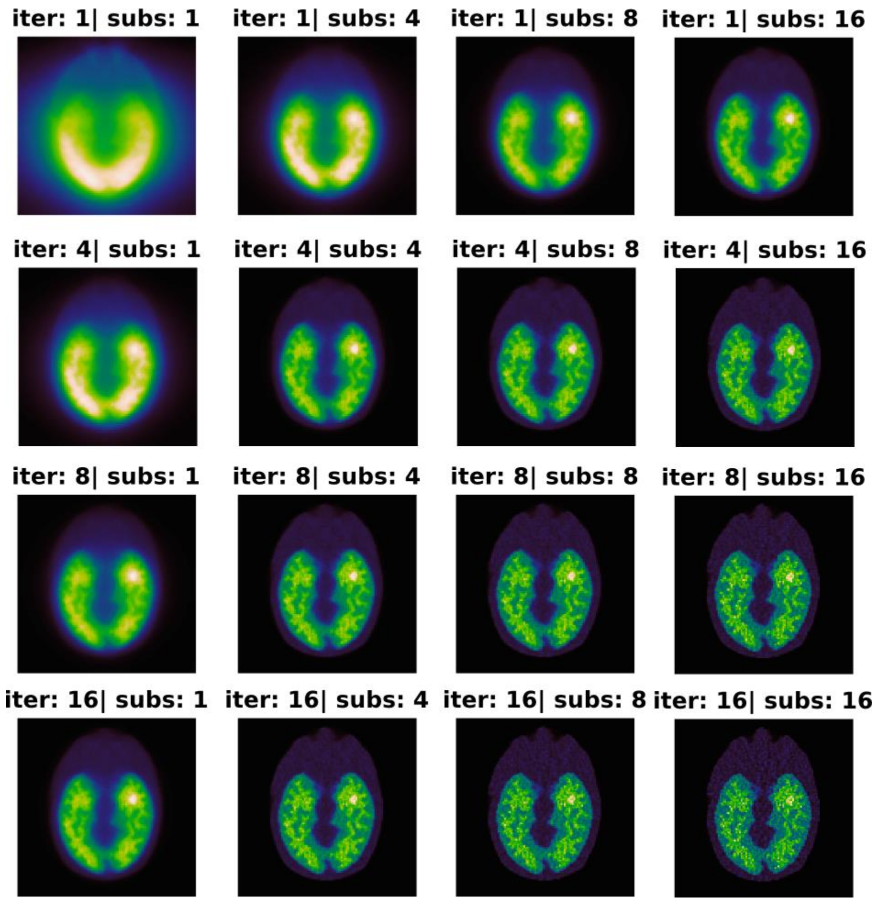
$$\left.\hat{f}^{(n+1)} = \frac{\hat{f}^{(n)}}{H^T 1} H^T \frac{g}{att.* \left(H\hat{f}^{(n)}\right) + scatt + rand} \right\} \textbf{ML-EM}$$

$$\textbf{OS-EM} \left\{ \hat{f}_k^{(n+1)} = \frac{\hat{f}_k^{(n)}}{H_k^{T} 1_k} H_k^{T} \frac{g_k}{att_k.* \left(H_k \hat{f}_k^{(n)}\right) + scatt_k + rand_k} \right.$$

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 (5)

5. Valutazione dell'accelarazione 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 vericare che ci sia una velocizzazione del processo di ricostruzione a parita di iterazioni utilizzate.

NOTA BENE:

- mettersi nella condizione iter_mlem = iter_osem*nblock
- vericare tutte e 3 le combinazioni possibili
- assicurarsi di misurare l'effettivo tempo di ricostruzione

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

Esercitazione (6)

FACOLTATIVO

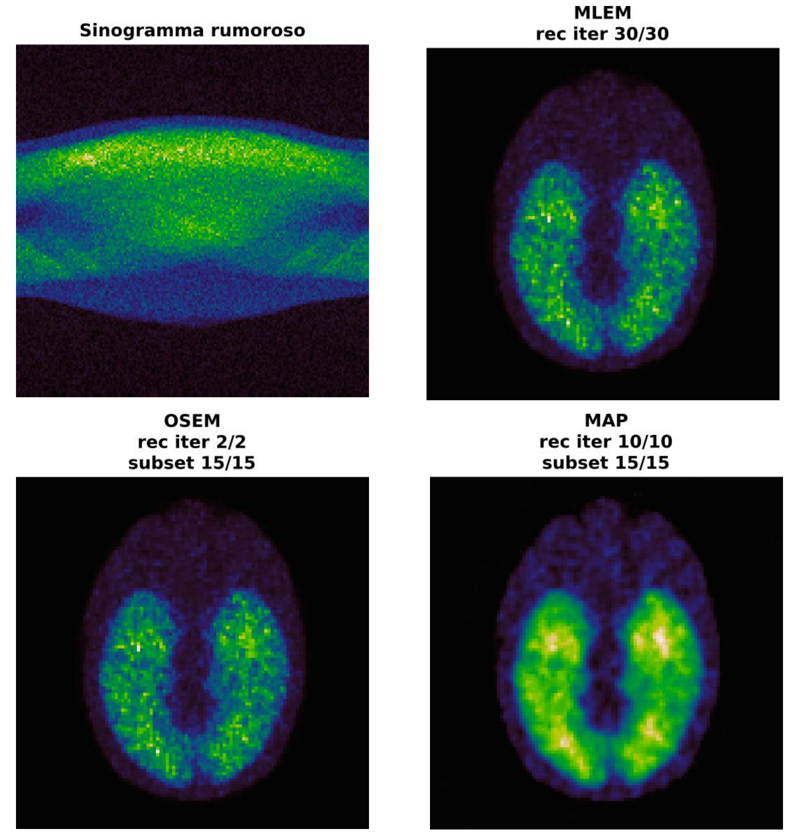
1. Ricostruzione MAP-OSL con prior quadratico

a) Partire (a scelta) dal codice MLEM o OSEM precedentemente implementati

b) Individuare il **punto** dell'algoritmo **in cui inserire il prior** facendo riferimento alle formule discusse a lezione

c) Il **kernel 3x3 fornito** è tale da calcolare **dE(f)/df** tramite una semplice **conv2**

d) verificare **come cambia il comportamento al variare di beta (peso del prior)** e come per valori di beta troppo alti il denominatore della MLEM diventa negativo e la ricostruzione non converge)



Ricostruzione MAP: approssimazione One Step Late

Log-likelihood Poissoniana con prior (log-likelihood penalizzata)

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

Exact MAP

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j} - \boldsymbol{\beta} \left.\frac{\boldsymbol{\partial E(f)}}{\boldsymbol{\partial f_j}}\right|_{\cancel{\hat{f}_j^{(n+1)}}}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}$$

One Step Late (OSL)

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j} - \boldsymbol{\beta} \left.\frac{\boldsymbol{\partial E(f)}}{\boldsymbol{\partial f_j}}\right|_{\hat{f}_j^{(n)}}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}}$$