UNIVERSITÀ DI PISA

SCUOLA DI INGEGNERIA

C.d.L. Magistrale in Ingegneria Biomedica

Corso di Immagini Biomediche

Ricostruzione iterativa di immagini tomografiche

Indice

1 Introduzione

Tradizionalmente, quello della ricostruzione di immagini tomografiche è stato percepito come un problema matematico relativamente semplice, basato sull'inversione della trasformata di Radon discreta. In quest'ottica, i dati sono trattati come integrali di linea dell'oggetto acquisito e non si tenta in alcun modo di modellare esplicitamente la randomicità tipica del processo di *conteggio* dei fotoni γ. Questa versione semplificata del problema di ricostruzione viene quindi risolta (in modo *esatto*) mediante *filtered backprojection* (FBP), un algoritmo che consente di generare immagini tomografiche in modo estremamente veloce. Sfortunamente, il modello FBP non è sufficiente a descrivere in modo esatto i dati reali di un'acquisizione di tomografia di emissione, per cui le immagini prodotte tendono a mostrare inaccuratezze significative.

Piuttosto che affidarsi al modello Radon, le moderne tecniche di ricostruzione fanno riferimento ad un più generico modello lineare, sufficientemente flessibile da consentire una descrizione estremamente dettagliata dei meccanismi di *blurring* ed attenuazione, così come delle altre sorgenti di artefatti che possono caratterizzare un'acquisizione reale. Oltre a questo, le tecniche statistiche di ricostruzione cercano di incorporare nel modello lineare anche una descrizione probabilistica del rumore presente nei dati misurati.

Il prezzo di questi miglioramenti ovviamente lo si paga con un modello matematico molto più complesso da risolvere della banale inversione della trasformata Radon. La soluzione in questo caso non può essere espressa in modo analitico o, anche quando ciò è possibile, risulta comunque impraticabile calcolarla con gli strumenti attualmente a disposizione. Di conseguenza, la maggior parte di queste tecniche sono sviluppate come *algoritmi iterativi*, puntando a raffinare progressivamente la stima dell'immagine ricostruita.

Il *trade-off* tra tecniche iterative ed FBP è quindi una scelta tra accuratezza ed efficienza della ricostruzione. Gli algoritmi iterativi, infatti, si trovano a dover calcolare ripetutamente operazioni di proiezione e retroproiezione, e di conseguenza il tempo computazionale è invariabilmente maggiore di quello richiesto dalla FBP. Inizialmente questo è stato un ostacolo verso l'adozione di questo tipo di tecniche nella pratica clinica. Un'ulteriore differenza tra metodi iterativi e FBP è nell'aspetto delle immagini prodotte, in particolare per quanto riguarda la rappresentazione delle "alte frequenze" dell'immagine, come rumore e dettagli: i medici che utilizzano tali ricostruzioni per fini diagnostici devono essere consapevoli delle differenze tra le due tecniche per poter valutare correttamente i risultati (motivo per cui non c'è mai stato un verdetto definitivo sulla superiorità dei metodi iterativi rispetto a quelli analitici, e i due approcci continuano a coesistere nella maggior parte degli scanner clinici).

In questa dispensa ci concentreremo sulla presentazione dei principi generali della ricostruzione iterativa. Inizieremo con la descrizione del problema tomografico come problema inverso lineare, definendo le caratteristiche statistiche dei dati misurati. Descriveremo le principali caratteristiche di una generico algoritmo iterativo;ed infine ci concentreremo sull'analisi di due degli algoritmi più diffusamente utilizzate attualmente, ossia la Maximum Likelihood Expectation Maximization (ML-EM d'ora in avanti), e la Ordered Subsets Expectation Maximization (OS-EM, d'ora in avanti).

2 Ricostruzione tomografica come problema lineare inverso

2.1 Un modello lineare per descrivere la creazione delle immagini

Per quanto riguarda la tomografia ad emissione, il problema di ricostruzione delle immagini può essere formulato nel modo seguente:

Stimare la distribuzione spaziale di tracciante f, dati (1) un insieme di proiezioni angolari misurate g, (2) delle informazioni (sotto forma di una matrice H) riguardo il sistema di acquisizione, e, se possibile, (3) una descrizione statistica dei dati e (4) una descrizione statistica dell'oggetto (Fig.1).

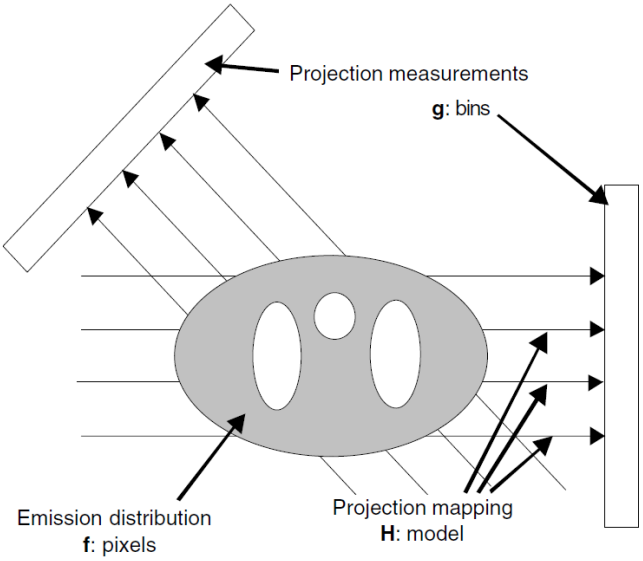


Figura 1: Modello generale di una proiezione tomografica in cui le misure sono date da integrali pesati della distribuzione dell'oggetto che emette fotoni

Questa definizione vale sia per la PET che per la SPECT, indipendentemente dal tipo di harware utilizzato. Nell'ipotesi di linearità del nostro problema, la ricostruzione si configura come un classico problema inverso del tipo:

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

dove $f(x)$ è un vettore definito nel dominio immagine, g_i rappresenta l'*i-esima* proiezione, e $h_i(x)$ è la risposta dell'*i-esima* misura alla sorgente x. Per acquisizione 2D (singola fetta tomografica), $D = 2$, per acquisizione 3D, $D = 3$. $h_i(x)$ può quindi essere vista come la *point spread function* del sistema di acquisizione, e può essere definita in modo da contenere anche info riguardo effetti di attenuazione e *blurring*. Dal modello in (1) sono stati omessi, per chiarezza i contributi additivi, come conteggi accidentali e dovuti a scattering, tipici della PET. Ciò che distingue la tomografia da altri problemi descrivibili dallo stesso modello lineare è che le g_i sono proiezioni.

Ai fini implementativi, l'immagine ricostruita non può essere rappresentate come una funzione definita su un dominio continuo. Piuttosto quello che otteniamo è una sorta di versione campionata (nello spazio) ed esprimibile nel dominio discreto come un vettore colonna f (Fig.2).

L'equazione (1) può quindi essere approssimata dal seguente sistema lineare di equazioni:

$$g_i = \mathbf{h}_i(x)^T\mathbf{f}, i = 1, \ldots, P \tag{2}$$

che in forma matriciale diventa:

$$\mathbf{g} = \mathbf{Hf} \tag{3}$$

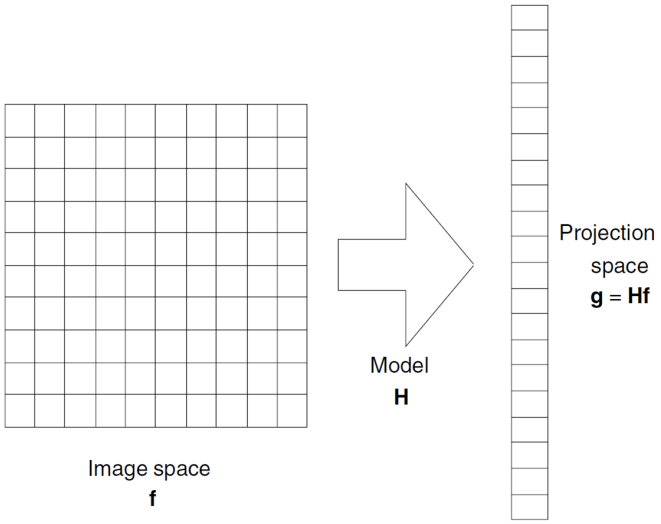


Figura 2: Modello discreto del processo di proiezione

Qui h_i è l'*i-esima* riga della matrice **H**, ed ogni elemento di **f**, ossia $f_j, j = 1, \ldots, N$, rappresenta un pixel nel dominio dell'immagine. Secondo questa notazione generale, **f** può rappresentare indifferentemente l'immagine di una fetta 2D o di un volume 3D, mentre la matrice **H**, opportunamente definita, riesce a modellare un apparato di imaging comunque complesso.

Anche lo spazio delle proiezioni è discreto, con i dati di proiezione rappresentati dal vettore **g**. Gli elementi di **g** sono chiamati *projection bins* o più semplicemente *bins*, ed ognuno di essi rappresenta i conteggi registrati rispetto ad una determinata LOR.

In eq.(3), **H** è una matrice $P \times N$ detta matrice di sistema e descrive l'intero processo di registrazione delle immagini. Ogni elemento di **H**, indicato come h_{ij}, rappresenta il contributo medio del pixel *j* dell'oggetto al valore del bin *i* del sinogramma. Definendo opportunamente **H** il modello del processo di proiezione può diventare tanto semplice quanto complesso, secondo le nostre esigenze, dal momento che l'intensità di un bin del sinogramma risulterà poi dalla somma pesata delle intensità dei pixel dell'immagine ad esso riferiti. Per rappresentare il caso descritto dalla trasformata Radon, gli elementi della matrice di sistema sono definiti in modo che un certo bin riceve contributo solo dai pixel che si trovano lungo una certa LOR, mentre il peso per tutti gli altri è nullo.

2.2 Modello statistico

Finora abbiamo discusso soltanto del comportamento *medio* del nostro sistema di imaging, trascurando quella che è l'intrinseca variabilità propria del processo di conteggio dei fotoni che è alla base di ogni acquisizione di tomografia ad emissione. Una riformulazione più corretta della (3) dovrebbe infatti essere:

$$E[\mathbf{g}] = \mathbf{Hf} \tag{4}$$

con $E[\cdot]$ che indica l'operazione di aspettazione.

L'emissione di fotoni obbedisce ad una distribuzione statistica di Poisson e, di conseguenza, anche la loro rilevazione da parte dello scanner, ammesso che il *dead time* del detettore possa essere trascurato e che nessun fattore correttivo sia stato applicato in fase di preprocessing dei dati. Il numero di eventi registrati in ciascun bin è indipendente da tutti gli altri. La legge di probabilità per **g** è data quindi da:

$$p(\mathbf{g}|\mathbf{f}) = \prod_{i=1}^{p} \frac{\bar{g}_i^{g_i} \exp(-\bar{g}_i)}{g_i!} \tag{5}$$

dove $\bar{g}_i$ è l'*i-esimo* elemento di $E[\mathbf{g}]$:

$$\bar{g}_i = \sum_{j=1}^{N} h_{ij} f_j \tag{6}$$

Il modello di Poisson (5) descrive bene i dati di emissione non corretti ed è il più utilizzato nel campo della tomografia ad emissione (PET e SPECT). Tuttavia, se il sistema di imaging corregge internamente i sinogrammi, la statistica muta. Solitamente si dovrebbe ricorrere ad un modello *Shifted Poisson*, ma più generalmente si opta per un modello *Gaussiano* ben più facilmente gestibile (d'altra parte, se il numero medi 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 $\bar{g}_i$).

3 Elementi di un algoritmo di ricostruzione iterativo

E' importante tener presente che ogni metodo di ricostruzione 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 è la strategia di ricostruzione, l'algoritmo la definizione dei singoli passi necessari ad implementare tale strategia.

4 Criteri di ottimizzazione

4.1 Ottimizzazione vincolata

Un approccio semplice alla ricostruzione consiste nel vedere il problema come la 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 (cfr. 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. Per questo motivo, nonostante queste tecniche abbiano goduto di un certo successo agli inizi dello sviluppo delle tecniche di tomografia ad emissione, attualmente sono stati quasi del tutto soppiantati da tecniche statistiche, basate sulla massima verosimiglianza e su approcci Bayesiani.

4.2 Criterio di massima verosimiglianza

Il criterio di massima verosimiglianza (*Maximum Likelihood*, ML) è un criterio di stima standard, proposto da R. A. Fisher (1921). L'ipotesi che si fa è che la legge di probabilità $p(\mathbf{g}|\mathbf{f})$, associata ad un vettore di osservazioni $\mathbf{g}$, sia definita rispetto ad un vettore incognito di parametri deterministi, $\mathbf{f}$,

che nel nostro caso è l'oggetto che vogliamo ricostruire nel dominio immagine. In questo contesto, $p(\mathbf{g}|\mathbf{f})$ è chiamata funzione di verosimiglianza (*likelihood*) e viene solitamente indicata come $L(\mathbf{f})$.

Il criterio di massima verosimiglianza si configura quindi come un'indicazione rispetto alla quale decidere quale immagine, tra tutte le immagini possibili, è la stima migliore dell'oggetto reale. Un definizione *formale* potrebbe essere la seguente:

> **Criterio ML**: scegliere come ricostruzione ottima l'immagine $\hat{\mathbf{f}}$ che ha la massima probabilità di aver generato i dati misurati $\mathbf{g}$.

Ciò significa che il criterio ML cerca la soluzione caratterizzata dalla maggiore vicinanza statistica alle osservazione misurate. Simbolicamente possiamo scrivere:

$$\hat{\mathbf{f}} = \arg\max_f p(\mathbf{g}|\mathbf{f}) \tag{7}$$

che signica scegliere il valore di $\mathbf{f}$ per cui $p(\mathbf{g}|\mathbf{f})$ è più grande. Lo stimatore ML ha alcune proprietà estremamente desiderabili che ne giustificano l'utilizzo in svariate occasioni:

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

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. 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 dell'aspettazione al valore vero. Come vedremo più avanti, lo *smoothing* può essere introdotto in due modi:

- **esplicitamente**: utilizzando filtri passa basso o regolarizzatori bayesiani;
- **implicitamente**: interrompendo prematuramente (prima di convergere all'effettiva soluzione ML) le iterazioni dell'algoritmo.

4.3 Criterio dei minimi quadrati

Nei problemi statistici di stima in cui la likelihood non è nota, una soluzione alternativa possibile è l'utilizzo di un approccio ai minimi quadrati (*least squares*, LS) per ricercare la soluzione migliore. Nel contesto della ricostruzione di immagini, il criterio LS può essere formulato come:

> **Criterio LS**: scegliere il valore di $\mathbf{f}$ che, se osservato attraverso la matrice di sistema $\mathbf{H}$, garantisce delle proiezioni $\mathbf{Hf}$ più possibile simili alle proiezioni osservate $\mathbf{g}$ (in termini di *distanza Euclidea*).

La soluzione LS pertanto punta a massimizzare la consistenza tra i dati osservati e l'immagine ricostruita. Simbolicamente possiamo scrivere:

$$\begin{aligned} \hat{\mathbf{f}} &= \arg\min_f \|\mathbf{g} - \mathbf{Hf}\|^2 \\ &= \arg\min_f \sum_{i=1}^{p} (g_i - \sum_{j=1}^{N} h_{ij} f_j)^2 \end{aligned} \tag{8}$$

L'equazione (8) può essere risolta analiticamente per ottenere una soluzione in forma chiusa del tipo:

$$\hat{\mathbf{f}} = \mathbf{H}^{+}\mathbf{g} = (\mathbf{H}^{T}\mathbf{H})^{-1}\mathbf{H}^{T}\mathbf{g} \tag{9}$$

dove $\mathbf{H}^{+}$ è detta *pseudoinversa* di **H** (assumendo che $\mathbf{H}^{T}\mathbf{H}$ sia invertibile). Questa soluzione in forma chiusa non è usata spesso nella tomografia ad emissione per via delle grandi dimensioni della matrice di sistema **H**, e spesso si preferisce ricorrere a soluzioni iterative.

Se abbiamo la possibilità di sapere che alcune delle proiezioni g_i hanno una varianza molto maggiore delle altre, possiamo pensare di pesare ciascuno dei termini di errore in (8) in modo non uniforme. Questo approccio prende il nome di stima *weighted-least-squares* (WLS) e può essere formulato come:

$$\begin{aligned}\hat{\mathbf{f}} &= \arg\min_{f}(\mathbf{g}-\mathbf{H}\mathbf{f})^{T}\mathbf{D}(\mathbf{g}-\mathbf{H}\mathbf{f}) \\ &= \arg\min_{f}\sum_{i=1}^{p} d_i(g_i - \sum_{j=1}^{N} h_{ij}f_j)^2\end{aligned} \tag{10}$$

con D matrice diagonale i cui elementi d_i sono generalmente scelti come $(var[g_i])^{-1}$. Nella tomografia ad emissione, i cui dati sono distribuiti secondo Poisson, la varianza è uguale alla media, per cui $d_i = (\bar{g}_i)^{-1}$. Come per la soluzione LS, anche quella WLS può essere espressa in forma chiusa, come:

$$\hat{\mathbf{f}} = (\mathbf{H}^{T}\mathbf{D}\mathbf{H})^{-1}\mathbf{H}^{T}\mathbf{D}^{-1}\mathbf{g} \tag{11}$$

ma, di nuovo, in genere si preferisce ricorrere ad una stima iterativa per via delle dimensioni di **H**. Un'altra possibile versione della stima WLS prevede di ricalcolare la matrice dei pesi **D** ad ogni iterazione, sulla base della proiezione dell'immagine stimata al passo precedente.

E' importante sottolineare che, sebbene gli stimatori LS e WLS non facciano esplicito riferimento a nessun modello probabilistico dei dati, all'atto pratico risultano equivalenti allo stimatore ML nell'ipotesi di distribuzione Gaussiana (basta confrontare la funzione WLS in (10) con l'espressione di una likelihood Gaussiana).

4.4 Limite intrinseco degli approcci ML, LS e WLS

Il problema principale che accomuna tutte le tecniche descritte precedentemente è che queste tendono *per definizione* a produrre delle immagini estremamente rumorose. Questo segue direttamente dalla loro tendenza a garantire la massima consistenza con i dati misurati: avendo a che fare con dati intrinsecamente rumorosi, l'immagine che è massimamente consistente con essi sarà per forza di cose rumorosa a sua volta.

Questo è particolarmente evidente nella tomografia ad emissione dato che i sistemi di imaging agiscono naturalmente come filtri passa-basso (tendono a produrre immagini sfocate), mentre il rumore ha uno spettro che si estende a tutte le frequenza. Consideriamo l'esempio LS. La soluzione è ottenuta calcolando la pseudoinversa di **H**. Se **H** agisce da passa-basso, la sua inversa sarà un operatore passa-alto che tenderà ad amplificare il rumore nell'immagine ricostruita. Se consideriamo il rumore come un disturbo puramente additivo ($\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}$):

$$\begin{aligned}\hat{\mathbf{f}} &= \mathbf{H}^{+}\mathbf{g} = \mathbf{H}^{+}(\mathbf{H}\mathbf{f}+\mathbf{n}) \\ &= \mathbf{f} + \mathbf{H}^{+}\mathbf{n}\end{aligned} \tag{12}$$

Il primo termine è la soluzione corretta, ma il secondo termine è puro rumore sottoposto ad un filtro passa-alto che ne amplifica le componenti ad alta frequenza, rendendo $\hat{\mathbf{f}}$ generalmente rumorosa.

4.5 Metodi Bayesiani

In genere si fa riferimento ai metodi ML, LS e WLS come criteri *classici* di stima, facendo riferimento all'assunzione di base secondo cui **f** è una grandezza incognita ma deterministica (non una variabile casuale). L'idea è che i dati, da soli, sono quindi in grado di determinare la soluzione e che nessun *pregiudizio*, nessuna ipotesi fatta da parte dello sperimentatore debba influenzare la stima.

In contrasto, i metodi Bayesiani partono dall'assunzione che la quantità incognita **f** sia essa stessa *random* e possa quindi essere descritta da una distribuzione di probabilità (PDF) $p(\mathbf{f})$ nota già prima (*a priori*) dell'acquisizione della misura. Questo ci consente di guidare il processo di stima attraverso la formulazione di ipotesi basate su ciò che ci si aspetta di trovare. Ad esempio, se acquisiamo l'immagine del cervello di un paziente, possiamo pensare di trattare il cervello di quella persona come un campione casuale estratto da una ipotetica popolazione di cervelli. La PDF $p(\mathbf{f})$ viene comunemente detta *prior* ed è ciò che ci consente di assumere che l'immagine del cervello che stiamo ricostruendo avrà, ad esempio, probabilità nulla di somigliare ad una macchina, piuttosto che ad un cuore, ed una probabilità positiva di somigliare ad un cervello.

Come facilmente immaginabile, esprimere queste conoscenze a priori in forma matematica è estremamente complesso ed in genere si opta per ipotesi meno ambiziose come quella di rafforzare l'ipotesi di *smoothness* dell'immagine finale ricostruita, in modo da ridurne il rumore: si associa bassa probabilità ad immagini ricche di dettagli fini, ipotizzando che nella maggior parte dei casi essi siano dovuti a rumore per via dell'assuzione fatta che il sistema di imaging **H** agisca naturalmente da filtro passa-basso.

In questa sede non scenderemo in dettaglio di questa classe di metodi. Giusto per mettere in forma simbolica quanto introdotto in questo paragrafo, ci basti dire che i metodi Bayesiani di ricostruzione sono in genere catalogati come approcci MAP (*Maximum a Posteriori*) in cui l'obiettivo è massimizzare la distribuzione a posteriori, $p(\mathbf{f}|\mathbf{g})$ che, dal teorema di Bayes, può essere espressa come:

$$\hat{\mathbf{f}} = \arg\max_f p(\mathbf{f}|\mathbf{g}) = \arg\max_f \frac{p(\mathbf{g}|\mathbf{f})p(\mathbf{f})}{p(\mathbf{g})} \tag{13}$$

Possiamo notare che calcolando il logaritmo della posterior e trascurando il termine $p(\mathbf{g})$ che non dipende da **f**, il criterio MAP si semplifica in:

$$\hat{\mathbf{f}} = \arg\max_f [\ln p(\mathbf{g}|\mathbf{f}) + \ln p(\mathbf{f})] \tag{14}$$

ottenendo una formulazione molto simile (dal punto di vista formale) ad un criterio ML (7) con l'aggiunta di un termine di penalizzazione dato dal logaritmo del *prior* scelto. Questo tipo di approccio penalizzato può ovviamente essere applicato anche agli stimatori ML, LS e WLS: ciò che differenzia l'approccio Bayesiano è nelle ipotesi probabilistiche che sono alla base della definizione dei *priors* stessi.

5 Algoritmi di ricostruzione iterativa

Molti approcci differenti sono stati proposti per risolvere il problema di ricostruzione di immagini di tomografia ad emissione. La maggior parte di essi condivide molte caratteristiche comuni. In

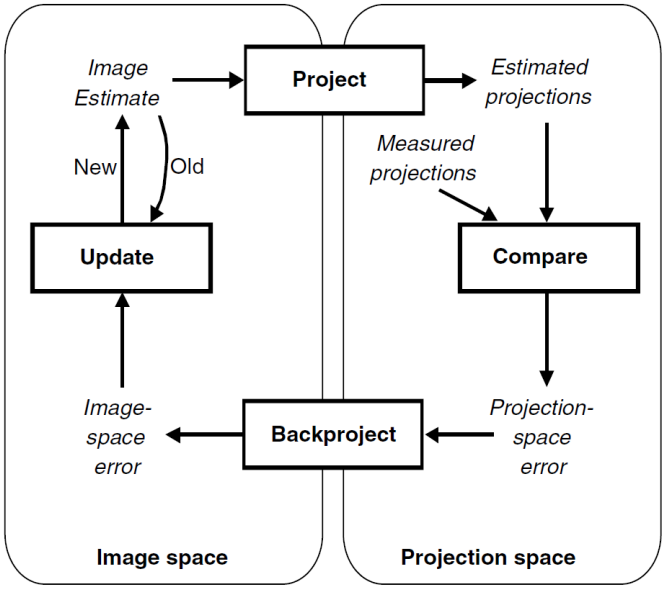


Figura 3: Diagramma di flusso di un generico algoritmo di ricostruzione iterativa

questa sezione discuteremo le caratteristiche comuni dei principali algoritmi di ricostruzione iterativa, presentandone le principali proprietà, per concentrarci poi maggiormente sull'approccio basato sull'ottimizzazione del criterio ML.

5.1 Struttura generale di un algoritmo iterativo

La maggior parte degli algoritmi iterativi di ricostruzione segue lo schema in fig. 3. Il processo inizia con una stima iniziale $\hat{\mathbf{f}}^{(0)}$ del valore di intensità dei pixel dell'immagine. Ad una generica iterazione n avremo quindi la stima corrente dell'immagine indicata come $\hat{\mathbf{f}}^{(n)}$. Questa stima intermedia viene proiettata nel dominio dei sinogrammi, generando un set di valori $\hat{\mathbf{g}}^{(n)}$. I valori di proiezione predetti $\hat{\mathbf{g}}^{(n)}$ vengono quindi confrontati con i valori misurati $\mathbf{g}$ producendo un vettore di errore $\mathbf{e}_g$ definito nello spazio proiettato. Questo è quindi rimappato nello spazio immagine attraverso un'operazione di retroproiezione per produrre un'immagine-errore $\mathbf{e}_f$ che viene usata per aggiornare la stima, $\hat{\mathbf{f}}^{(n+1)}$. Il tutto si ripete fino a che le iterazioni non si fermano in modo automatico (convergenza) o vengono interrotte. La stima corrispondente all'ultima iterazione effettuata viene trattata come soluzione finale.

Il modo in cui le operazioni di proiezione, confronto, retroproiezione ed aggiornamento sono realizzate, è per l'appunto ciò in cui i diversi algoritmi si distinguono tra loro. Osservando lo schema in fig.3 possiamo vedere la differenza principale con la ricostruzione analitica FBP: essa utilizza soltanto la porzione di retroproiezione dell'intero *loop*, e così facendo non gode di nessun feedback riguardo la consistenza dell'immagine stimata con i dati di proiezione misurati (ma ovviamente risulta estremamente più veloce).

5.2 Algoritmi di ottimizzazione vincolata (ART)

Sono la prima classe di algoritmi iterativi sviluppati, basati sulla strategia accennata in (4.1) e progettati per rislvere sistemi lineari di equazioni. Nel contesto della tomografia ad emissione questi metodi sono generalmente indicati come *tecniche di ricostruzione algebrica* (ART) ed esistono in molte varianti.

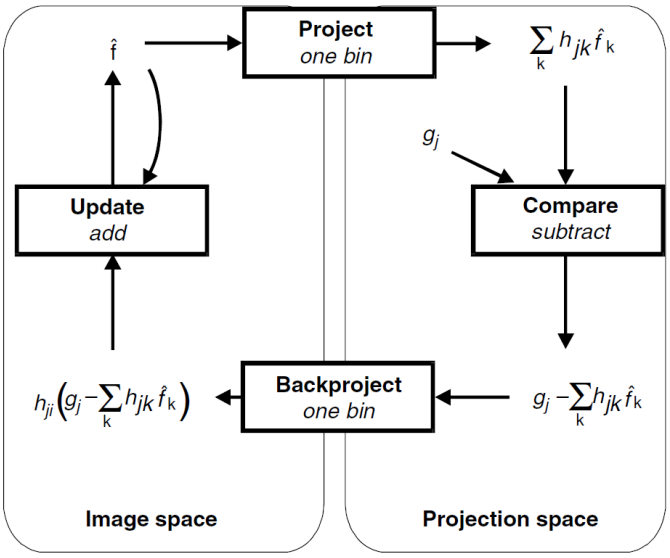


Figura 4: Algebraic Reconstruction Tecnique (ART)

Questa classe di tecniche è già stata affrontata in un'altra lezione per cui non verrà approfondita in questa sede. Attualmente gli algoritmi ART sono molto poco utilizzati nella tomografia ad emissione ma sono comunque utili per capire i meccanismi classici che entrano in gioco nella soluzione dei problemi di ricostruzione iterativa

Il modello $\mathbf{g} = \mathbf{Hf}$ viene trattato come un insieme di equazioni lineari, ognuna associata ad un bin del sinogramma. Ogni equazione lineare, $g_i = \mathbf{h}_i^T \mathbf{f}$ definisce un *iperpiano* nello spazio vettoriale in cui è definita $\mathbf{f}$. Ipotizzando che il sistema di equazioni sia consistente (i.e. non c'è rumore nelle misure) la soluzione è costituita da tutti i punti che giacciono all'intersezione di tutti gli iperpiani.

Partendo da una stima iniziale $\hat{\mathbf{f}}^{(0)}$, questa viene proiettata (nel senso algebrico del termine, non tomografico!) su tutti gli iperpiani ripetutamente (fig.5)

La regola di update più comune per l'ART è:

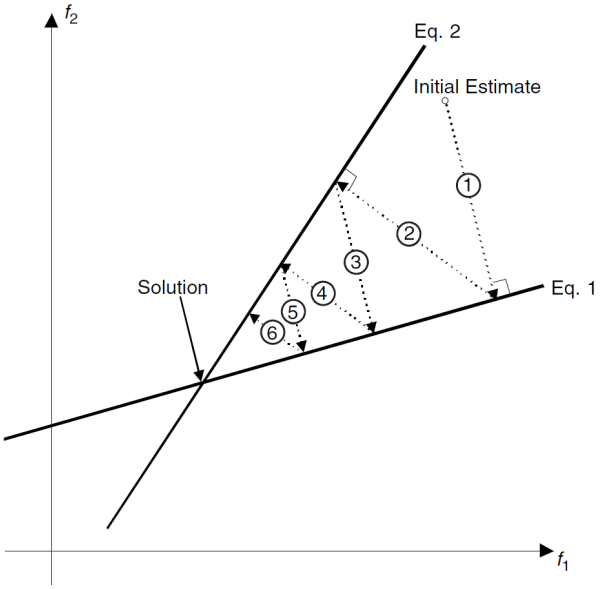


Figura 5: Semplice esempio dell'algoritmo ART in presenza di due soli pixel (2 dimensioni) e 2 misure (2 set di vincoli)

$$\hat{f}_i^{(n+1)} = \hat{f}_i^{(n)} + h_{ji}\frac{(g_j - \sum_k h_{jk}\hat{f}_k^{(n)})}{\sum_k h_{jk}^2} \tag{15}$$

Tuttavia molte altre versioni ne esistono, che introducono vincoli di non-negatività, o che piuttosto riformulano il problema di update come moltiplicativo (Multiplative ART). Un'altra versione che ha avuto un discreto successo è stata poi la Simultaneous Iterative Reconstruction Technique (SIRT).

5.3 Maximum-Likelihood Expectation-Maximization (MLEM)

Quando applicata al campo della tomografia ad emissione (o più precisamente a qualsiasi problema inverso lineare caratterizzato da rumore Poissoniano) il framework ML-EM (che trova applicazione in moltissimi campi, dei più diversi) porta alla seguente semplice equazione iterativa, facile da implementare e da capire:

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik}\hat{f}_k^{(n)}} \tag{16}$$

La formula (16) rappresenta il modello iterativo specificamente sviluppato da Lange e Carson (1984) per i problemi di ricostruzione di tomografia ad emissione. In realtà questa è solo un'applicazione specifica di un approccio generico allo sviluppo di procedure iterative per la soluzione di qualsiasi problema formulato rispetto ad un criterio ML. L'idea di base è che esistono moltissimi problemi per i quali la soluzione ML è estremamente difficile da trovare con gli strumenti a disposizione, ma allo stesso tempo sarebbe molto semplice se si disponesse di dati aggiuntivi a cui solitamente ci si riferisce chiamandoli *dati mancanti*.

A volte questi possono essere misure effettivamente mancanti (ad esempio, per problemi durante l'acquisizione). Molto più spesso la nozione di dati mancanti è un'astrazione che viene fatta con l'obiettivo di risolvere il problema in modo più conveniente: in questi casi i dati mancanti sono dati che sarebbe estremamente comodo avere a disposizione, ma mancano perchè non sono effettivamente misurabili in alcun modo. Questo è il caso con la tomografia ad emissione. Quando andiamo ad

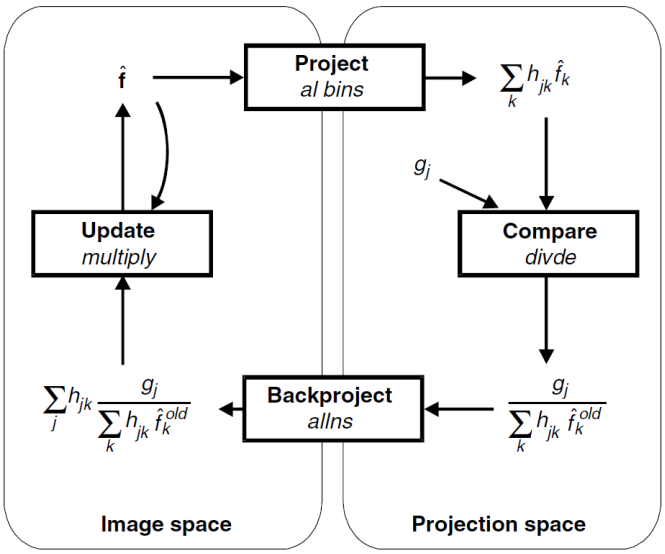


Figura 6: Maximum-Likelihood Expectation-Maximization

applicare un approccio EM, i dati osservati che sono stati effettivamente misurati, **g**, sono ipotizzati *incompleti* ed ipotizziamo l'esistenza di un set di dati *completo*, che chiamiamo **s**.

Il motivo per cui i nostri dati sono considerati incompleti è che non abbiamo nessuna idea di quale sia il punto di origine reale dei fotoni registrati da un certo bin del sinogramma. Il processo di proiezione tomografica mescola insieme i fotoni emessi dall'oggetto **f** all'interno del bin g_j attraverso un mixing lineare descritto dalla matrice di sistema **H**. L'obiettivo della ricostruzione tomografica è, quindi, quello di invertire l'effetto di **H** e separare i conteggi dei fotoni in base alla loro origine nello spazio immagine. Da questo punto di vista i nostri dati sarebbero completi se, oltre a sapere quanti conteggi sono stati registrati da bin *j*, sapessimo anche quanti di questi sono arrivati lì partendo dal pixel *k*. Un elemento del nostro set completo di dati sarà pertanto una quantità s_{jk} che indica il numero di fotoni emessi dal pixel *k* e rilevati dal bin *j*. Ovviamente, se già avessimo accesso in partenza a tale informazione, l'operazione di ricostruzione sarebbe triviale. Il motivo per cui ha senso ipotizzare l'esistenza di questo irrealistico datset completo è che far questo ci consente di usare un approccio EM per ricercare iterativamente la soluzione ML.

5.3.1 Definizione dell'algoritmo EM

L'algoritmo EM si compone di due passaggi alternati, che si ripetono fino a convergenza. Questi passaggi sono chiamati *aspettazione* (*Expectation*, E-step) e *massimizzazione* (*Maximization*, M-step) e sono definiti come segue:

E-step Partendo dalla stima attuale di $\hat{\mathbf{f}}^{(n)}$, si calcola:
$Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)}) = E[\ln p(\mathbf{s}|\mathbf{f})|\mathbf{g}, \hat{\mathbf{f}}^{(n)}]$

M-step La nuova stima $\hat{\mathbf{f}}^{(n+1)}$ è scelta come quella che massimizza $Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)})$:
$\hat{\mathbf{f}}^{(n+1)} = \arg\max_f Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)})$

L'essenza dell'algoritmo può essere spiegata come segue. In assenza di dati completi, la funzione log-likelihood relativa al set completo non può ovviamente essere calcolata. Ad ogni iterazione, pertanto, si calcola un'aspettazione di questa log-likelihood (E-step) e successivamente la si massimizza (M-step). E' possibile dimostrare (non verrà fatto in questa sede) che questa procedura consente di massimizzare anche la likelihood $p(\mathbf{g}|\mathbf{f})$, convergendo quindi verso la soluzione del problema ML originale.

5.3.2 Algoritmo EM per la tomografia di emissione

Vediamo ora come si arriva all'equazione (16). Richiamiamo che nel nostro caso un buona definizione di dato completo è s_{im}, che indica il numero (casuale) di fotoni emessi dal pixel *m* e rilevati dal bin *i*. I dati completi possono essere correlati alle proiezioni osservate **g** e all'immagine **f** nel modo seguente:

$$g_i = \sum_m s_{im} \tag{17}$$

$$E[s_{im}] = h_{im} f_m \tag{18}$$

L'E-step richiede un'espressione per la log-likelihood dei dati completi $\ln p(\mathbf{s}|\mathbf{f})$. Nella tomografia di emissione, i conteggi s_{im} sono variabili casuali distribuite secondo Poisson. Quindi:

$$p(\mathbf{s}|\mathbf{f}) = \prod_i \prod_m \frac{E[s_{im}]^{s_{im}} e^{-E[s_{im}]}}{s_{im}!} \tag{19}$$

mentre la log-likelihood vale (tenendo conto anche della (18)):

$$\ln p(\mathbf{s}|\mathbf{f}) = \sum_i \sum_m [s_{im} \ln(h_{im} f_m) - h_{im} f_m - \ln(s_{im}!)] \tag{20}$$

Ora abbiamo tutto quello che ci serve per calcolare l'E-step dell'algoritmo EM.

E-step

$$\begin{aligned} Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)}) &= E[\ln p(\mathbf{s}|\mathbf{f})|\mathbf{g}, \hat{\mathbf{f}}^{(n)}] \\ &= \sum_i \sum_m \{E[s_{im}|\mathbf{g}, \hat{\mathbf{f}}^{(n)}] \ln(h_{im} f_m) - h_{im} f_m - E[\ln(s_{im}!)]\} \end{aligned} \tag{21}$$

L'aspettazione condizionata di s_{im} in eq.(21) è data da:

$$E[s_{im}|\mathbf{g}, \hat{\mathbf{f}}^{(n)}] = g_i \frac{h_{im} \hat{f}_m^{(n)}}{\sum_m h_{im} \hat{f}_m^{(n)}} \equiv p_{im} \tag{22}$$

che rappresenta la frazione di conteggi rilevati nel bin *i* e che ci si aspetta provengano dal pixel *m*, nell'ipotesi che la stima corrente dell'immagine $\hat{\mathbf{f}}^{(n)}$ sia la sorgente effettiva di tali conteggi. Sostituendo (22) in (21):

$$Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)}) = \sum_j \sum_k \{p_{im} \ln(h_{im} f_m) - h_{im} f_m - E[\ln(s_{im}!)]\} \tag{23}$$

M-step Nell'M-step andiamo a cercare la nuova immagine $\hat{\mathbf{f}}^{(n+1)}$ massimizzando $Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)})$ rispetto ad $\mathbf{f}$. Per farlo è sufficiente mandare a zero la derivata di $Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)})$ e risolvere per $\hat{\mathbf{f}}^{(n+1)}$:

$$\begin{aligned} \frac{\partial Q(\mathbf{f}|\hat{\mathbf{f}}^{(n)})}{\partial f_j} &= 0 \\ &= \sum_i \left(\frac{p_{ij}}{\hat{f}_j^{(n+1)}} - h_{ij}\right) \end{aligned} \tag{24}$$

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}} \tag{25}$$

5.3.3 Proprietà della ML-EM

L'algoritmo ML-EM presentato nelle equazioni (16) e (25) ha pertanto una forma semplice che si adatta alla descrizione generica di algoritmo iterativo data precedentemente (cfr. fig.6). La principale differenza rispetto alla ART (in particolare rispetto alla forma moltiplicativa, MART, non trattata in questa sede) è che in questo caso tutti i pixel dell'immagine sono aggiornati simultaneamente. Dal momento che errore e aggiornamento sono termini moltiplicativi, l'algoritmo ML-EM impone di default un vincolo di non-negatività e consente di fissare a zero fin dall'inizio il valore di alcuni pixel.

I limiti, per quanto riguarda l'applicazione particolare alla tomografia di emissione, sono due. In primo luogo la convergenza dell'algoritmo, anche se garantita e predicibile, è lenta. Una soluzione che sia anche solo utilizzabile in genere richiede tra le 30 e le 50 iterazioni. Dato che ogni iterazioni richiede passaggi di proiezione e retro-proiezione, possiamo aspettarci dei tempi di esecuzione che

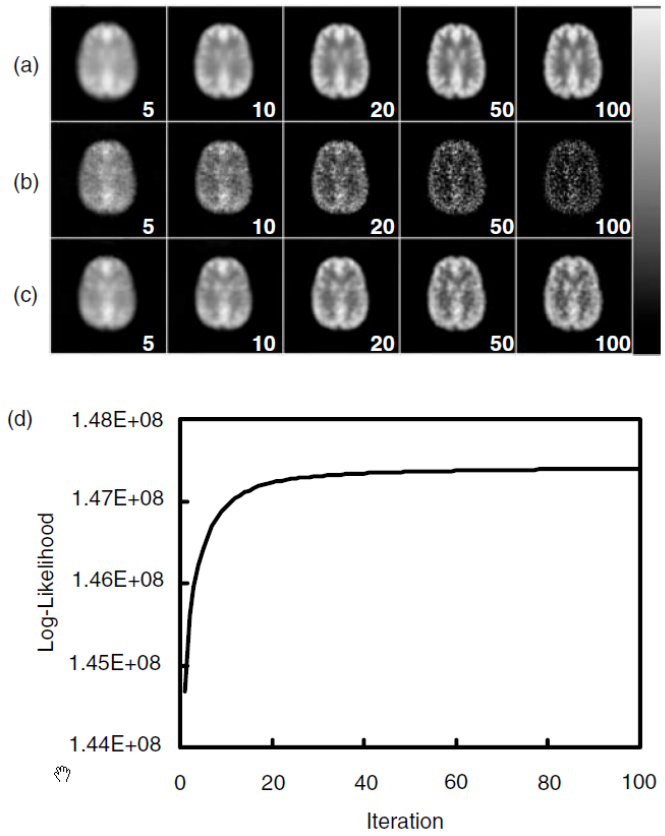


Figura 7: Proprietà di convergenza della ML-EM. Le immagini mostrano il risultato della stima iterativa nella ricostruzione di un fantoccio simulato. (a) dati non rumorosi; (b) dati rumorosi e (c) dati rumorosi con applicazione di un filtro di post-processing. (d) Il grafico mostra l'andamento della log-likelihood in funzione delle iterazioni, per il dataset rumoroso.

siano di circa uno o due ordini di grandezza superiori rispetto alla FBP. Se teniamo in considerazione il risultato significativamente migliore che si ottiene con questa metodica (capace di risolvere il generico modello lineare e quindi compensare per attenuazione non uniforme ed atri tipi di artefatti), possiamo tuttavia intuire perchè la ML-EM alla fine abbia prevalso lo scetticismo iniziale legato ai lunghi tempi di elaborazione.

Il secondo limite riguarda il criterio ML su cui si basa e che tende a produrre delle immagini estremamente rumorose (12). Al procedere delle iterazioni, l'algoritmo si avvicina alla soluzione ML e la varianza delle immagini stimate,che si manifesta sotto forma di rumore, cresce. Questo significa che la ML-EM riesce a garantire dei buoni risultati se la procedura iterativa viene interrotta prematuramente. Non è raro, inoltre, per l'applicazione clinica applicare al risultato finale un filtro passa-basso o un'interpolazione dell'immagine ricostruita.

Il comportamento dell'algoritmo ML-EM in funzione delle iterazioni è mostrato in fig. 7. E' stato più volte discusso e dimostrato che la ML-EM tende a far comparire prima le frequenze spaziali più basse, per poi sviluppare gradualmente le componenti ad alta frequenza spaziale al progredire delle iterazioni. Come intuibile dalla figura 7a, interrompere prematuramente l'iterazioni equivale quindi implicitamente ad un filtraggio passa-basso dell'immagine ricostruita.

Il grafico 7d mostra l'andamento della log-likelihood con le iterazioni, indicando come, nonstante si raggiunga molto presto una sorta di *plateau*, in realtà la stima dell'immagine continua a cambiare. Ne consegue che in genere la log-likelihood non è un buon indice di qualità dell'immagine finale per la stessa ragione per cui il criterio ML non è un buon criterio di convergenza. Inoltre immagini

caratterizzate dalla stessa log-likelihood possono in generale apparire estremamente diverse tra loro.

5.4 Ordered-Subset Expectation-Maximization (OS-EM)

Per affrontare il problema della lenta convergenza della ML-EM sono stati proposti, negli anni, diversi metodi di accelerazione, che in genere si sono concentrati sull'accelerare la convergenza iniziale, ossia sul raggiungere il *plateau* nel minor numero possibile di iterazioni. Spesso, una volta raggiunta questa condizione, la velocità di convergenza non migliorava poi di molto. Un'eccezione è stata la *ordered subset EM*, spesso indicata anche come *block-iterative reconstruction*.

L'algoritmo OS-EM (Hudson and Larkin, 1994) è concettualmente una semplice modifica della ML-EM, data da:

$$\hat{f}_j^{new} = \frac{\hat{f}_j^{old}}{\sum_{i' \in S_n} h_{i'j}} \sum_{i \in S_n} h_{ij} \frac{g_i}{\sum_{i=1}^{N} h_{il} \hat{f}_l^{old}} \tag{26}$$

In pratica, la retroproiezione viene applicata soltanto ai bin che appartengono al sottoinsieme (*subset*) S_n del sinogramma. Ad ogni aggiornamento, un diverso subset viene preso in considerazione in quella che viene definita *sottoiterazione* dell'algoritmo (mentre un intero passaggio di tutti i subset costituisce un'iterazione vera e propria). Il tempo di esecuzione di un'iterazione OS-EM è pertanto comparabile a quello ML-EM.

L'organizzazione dei subset è importante per le performance dell'algoritmo, Inoltre, se in qualcuno dei subset non compare il contributo di *tutti* i pixel del *field of view* (FOV) la prima sommatoria al denominatore della (26) è zero, ed evidenti problemi matematici possono insorgere.

Volendolo confrontare con altri algoritmi discussi precedentemente abbiamo due estremi opposti. Se utilizziamo un singolo subset che comprende tutti i bin di proiezione, riduciamo la OS-EM ad

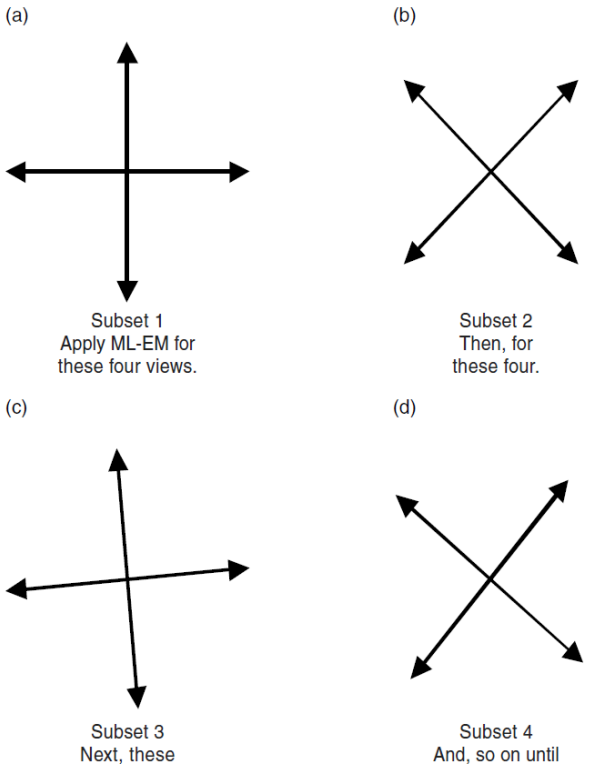


Figura 8: Sequenza di utilizzo dei vari subset.

una ML-EM. Se, invece, definiamo ogni subset come una singola proiezione, otteniamo un algoritmo simile alla ART moltiplicativa (ma in questo caso, per quanto detto poco sopra, non possiamo utilizzare un modello di **H** basato sulla struttura dei proiettori).

Solitamente, i subset sono organizzati in gruppi di proiezioni associate con un certo *punto di vista*. La figura (8) mostra una procedura tipica. In genere, i componenti di un subset sono scelti in modo da avere la massima distanza angolare tra loro. Ad esempio, se vogliamo creare 16 subset da un sinogramma costituito da 128 proiezioni che coprono 360°di arco di acquisizione, ogni subset conterrà tutti i bin di 8 *viste*, selezionate con intervallo angolare di 45°. Anche la successione con cui vengono processati i vari subset segue una logica analoga. Ad esempio, partiamo con un subset definito rispetto alle direzioni cardinali; il secondo avrà un incremento angolare di 22.5°dal primo; mentre il terzo tornerà ad avere una distanza dal primo di circa 2.8°(una distanza angolare, che è la massima rispetto al secondo subset, senza tornare a sovrapporsi al primo). E così via.

Come facilmente intuibile, il numero di subset scelti determina il grado di accelerazione rispetto alla ML-EM. Un legge empirica afferma che dopo *n* iterazioni di OS-EM si raggiunge un punto di convergenza paragonabile a quello ottenuto con (Numero di subset) x *n* iterazioni ML-EM. Ne consegue che anche il rumore cresca ad una velocità comparabile e che quindi l'algoritmo vada fermato ancora più precocemente, eventualmente applicando un post-filtraggio come per la ML-EM.

Nonostante tutte le somiglianza con la ML-EM, è infine importante sottolineare che la OS-EM non è *realmente* un algoritmo EM e quindi la sua convergenza non può essere dimostrata per via teorica. Ad oggi l'esperienza d'uso è stata buona e si è mostrata una tendenza a produrre dei risultati in genere quasi identici (nella media) a quelli della ML-EM. Il costo principale dell'accelerazione prodotta sta in un aumento del rumore (varianza) a partià di bias.

5.5 Least Squares (LS) e Weighted Least Squares (WLS)

I criteri LS e WLS introdotti nella sezione 4 conducono alla necessità di ottimizzare un funzione quadratica. Problemi di ottimizzazione che coinvolgono funzioni quadratiche sono stati studiati diffusamente in diversi ambiti applicativi; di conseguenza esistono moltissimi approcci diversi che possono essere sfruttati ai fini della ricostruzione WLS. Inoltre il caso LS è fondamentalmente un caso speciale del WLS in cui la matrice dei pesi è scelta come *identità*. Per cui tutti i metodi validi per WLS lo sono anche per LS.

I metodi che possono essere utilizzati per risolvere l'ottimizzazione WLS presentata in (10) condividono una forma base comune che si configura come un algoritmo di aggiornamento additivo:

$$\hat{f}^{(n+1)} = \hat{f}^{(n)} + t\Delta f^{(n)} \tag{27}$$

dove *n* è il numero di iterazione; *t* è uno scalare che rappresenta lo *step size* di aggiornamento; e il vettore $\Delta f^{(n)}$, che ha la stessa dimensione dell'immagine, rappresenta la *direzione di aggiornamento* all'interno dello spazio delle possibili soluzioni.

I diversi algoritmi si distinguono principalmente per come viene calcolata questa direzione di aggiornamento. I principali sono:

- **Gradient Descent**: utilizza il gradiente della funzione costo come direzione di update.
- **Coniugate Gradiente**: tutte le direzioni sono scelte tra loro coniugate, per cui la minimizzazione condotta in una direzione non influenza quella calcolata lungo le altre.
- **Coordinate Descent**: $\Delta f^{(n)}$ è un vettore con tutti gli elementi a 0, eccetto quello relativo al pixel corrente; in altre parole ogni iterazione aggiorna un pixel alla volta.

5.6 Maximum A Posteriori Reconstruction (MAP)

Abbiamo visto che generalmente la soluzione ML è troppo rumorosa per poter essere effettivamente utilizzata. Di conseguenza l'approccio standard pervede una combinazione di interruzione precoce delle iterazioni (prima della convergenza) e/o un filtraggio lineare passa-basso. Gli approcci MAP offrono, da questo punto di vista un approccio più flessibile e robusto (dal punto di vista teorico) per incoraggiare determinate caratteristiche nell'immagine ricostruita.

Data le somiglianza tra le funzioni costo MAP e ML, la maggior parte degli algoritmi di ricostruzione ML ha una controparte MAP. L'analogo della ML-EM è, dunque, la MAP-EM (Green, 1990):

$$\hat{f}_j^{(n+1)} = \frac{\hat{f}_j^{(n)}}{\sum_{i'} h_{i'j} + \beta \frac{\delta U(\mathbf{f})}{\delta \mathbf{f}}} \sum_i h_{ij} \frac{g_i}{\sum_k h_{ik} \hat{f}_k^{(n)}} \tag{28}$$

Questo algoritmo si distingue dalla ML-EM principalmente per l'aggiunta del *prior* a denominatore. Questo termine è problematico in quanto spesso per essere calcolato richiede la conoscenza di $\hat{f}_j^{(n+1)}$ che ovviamente non è ancora disponibile. Gli algoritmi MAP-EM differiscono appunto per il modo in cui affrontano questo problema.

L'approccio più comune consiste nel valutare il termine derivativo rispetto all'immagine stimata all'iterazione precedente. Questo approccio viene detto *One-Step-Late* (OSL). Molte altre versioni, più o meno complesse, sono state proposte a partire dalla seconda metà degli anni 90, ma non verranno approfondite in questa sede. In generale, la ricostruzione MAP punta ad alleviare i due principali problemi degli algoritmi ML. Primo, le ricostruzioni MAP sono meno rumorose delle controparti ML. Secondo, le iterazioni MAP tendono a raggiungere una condizione in cui variano molto poco da un'iterazione all'altra, indicando una approssimativa convergenza.

Il valore del parametro β influenza il peso che viene dato al prio di smoothing: valori bassi implicano un maggior dettaglio, ma anche un maggior rumore nelle immagini ricostruite; valori alti un rumore molto ridotto ma anche il rischio di perdere dettagli e contrasto. Il tutto ovviamente dipende anche molto dalla scelta del prior $U(\mathbf{f})$.

6 Conclusioni

In questa lezione abbiamo introdotto un discreto numero di approcci differenti alla ricostruzione di immagini, e ciò porta naturalmente alla domanda: quale di questi è il migliore? Sfortunamente, non esiste una risposta semplice: ogni algoritmo ha i suoi punti di forza e di debolezza, e può comportarsi in modo diverso in applicazioni diverse.

Abbiamo visto che gli algoritmi di ricostruzione iterativa condividono parecchi tratti comuni. Molti sono sufficientemente generici da poter essere applicati alla soluzione di ogni problema tomografico esprimibile come relazione lineare tra pixel dell'immagine e bin del sinogramma. La maggior parte può essere ben descritta da un modello generale che prevedere la ripetizione del processo di proiezione di una stima intermedia dell'immagine, il confronto della proiezione stimata con i dati misurati per calcolare un qualche indice di errore, e la retroproiezione dell'errore in modo da poterlo usare come aggiornamento della stima. Questo meccanismo di *feedback* garantisce che al crescere delle iterazioni la stima si avvicini all'immagine desiderata.

Dal momento che i dati misurati con la tomografia ad emissione sono random, anche l'immagine ricostruita da essi è random (ogni volta che si scansiona un oggetto si ottiene un'immagine leggermente differente). Prendendo in prestito alcuni strumenti della teoria della stima, un modo in cui

possiamo valutare quantitativamente il risultato della ricostruzione passa per il calcolo della media e della varianza dell'intensità rilevata in una certa regione di interesse.

Idealmente vorremmo che tale media fosse identica al valore vero. Il nostro successo da questo punto di vista può essere quantificato mediante il calcolo del *bias*, ossia della differenza tra immagine ricostruita e immagine vera, $b = E[\hat{\mathbf{f}}] - \mathbf{f}$. Vogliamo inoltre che la varianza della ricostruzione sia nulla, ossia che la ricostruzione sia esattamente la stessa, per ogni set di proiezioni rumorose dello stesso oggetto.

Ovviamente non è possibile raggiungere i due obiettivi simultaneamente. Ciò che accade è che gli sforzi per avvinarsi ad un *bias* nullo vanno ad incrementare la viarianza, e viceversa. Questo fenomeno è estremamente noto e solitamente si indica come *bias-variance-tradeoff*. Possiamo ridurre la varianza filtrando le immagini ricostruite fino al punto che non hanno più nessun dettaglio visibile. Chiaramente, il costo per questa bassissima varianza è un *bias* altissimo. Un *bias* ridotto può essere ottenuto non filtrando l'immagine ricostruita e lasciando l'algoritmo raggiungere la sua naturale convergenza (senza interromperlo prematuramente), ma così facendo l'immagine risultante mostrebbe una varianza molto alta (e apparirebbe come estremamente rumorosa).

Nella ricostruzione di immagini, il *trade of* tra bias e varianza può essere gestito agendo sul livello di smoothing applicato. Questo può essere controllato implicitamente (agendo sul numero di iterazioni della ML-EM/OS-EM) o esplicitamente (settando opportunamente il parametro β della MAP-EM). In entrambi i casi è possibile definire una curva bias-varianza spaziando in un range scelto per il parametro di controllo. L'obiettivo finale è puntare al miglior compromesso tra questi due fattori concorrenti.

7 Esercitazione

ALGORITMI ITERATIVI DI RICOSTRUZIONE DELLE IMMAGINI DI TOMOGRAFIA AD EMISSIONE

Maximum Likelihood Expectation Maximization (MLEM)
Ordered Subset Expectation Maximization (OSEM)

MATERIALE

- Fantoccio cerebrale 2D (111x111 pixel) preparato come file **brain.mat**
- Funzione *MATLAB* **Calcolo_A.m** già utilizzata nell'esercitazione sulle tecniche di ricostruzione analitiche
- Scheletro dell'esercitazione da svolgere con:
  1. predisposizione dei parametri base (liberamente modificabili) per ottenere risultati comparabili con quelli qui proposti;
  2. implementazione (già sviluppata) della parte di simulazione del sinogramma e delle sorgenti artefattuali di rumore (attenuazione, scattering e conteggi random) come già svolto in una delle esercitazioni precedenti;
  3. descrizione del codice da implementare e dei risultati da visualizzare.
- Scheletro della funzione **Calcolo_Hblock.m** con descrizione di quale debba essere la sua struttura e il suo funzionamento.

PUNTI DA SVOLGERE

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).
2. **Ricostruzione OSEM: calcolo dei blocchi della matrice di sistema**
   (a) Creare una funzione esterna (partire dal file Calcolo_Ablock.m fornito) per l'estrazione dei blocchi della matrice di sistema con cui ricostruire i singoli subset del sinogramma.
   (b) La descrizione della funzione è 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 è associato un determinato blocco.

3. **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_osem*nblock).

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

   (a) 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.

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

   (a) 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 iter_mlem = iter_osem*nblock
   - verificare tutte e 3 le combinazioni possibili (es. iter_mlem=40; iter_osem=2 nblock=20; iter_osem=20 nblock=2)
   - assicurarsi di misurare l'effettivo tempo di ricostruzione, privo di sovraccarichi legati a salvataggio di ricostruzioni intermedie e visualizzazione degli stessi

FACOLTATIVO

1. **Ricostruzione MAP-OSL con prior quadratico - smoothing -**

   (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 (attenzione a cosa è 'f', quale dimensione ha normalmente, quale dimensione deve avere per poter essere convoluta con il kernel, e se è necessario fare un reshape prima di inserire il prior stimato nell'algoritmo di ricostruzione)

   (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 più)

RISULTATI ATTESI

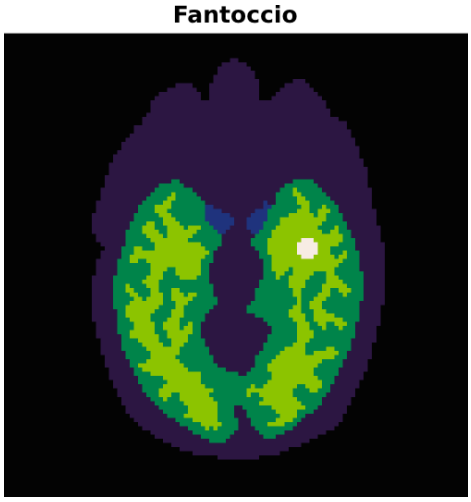
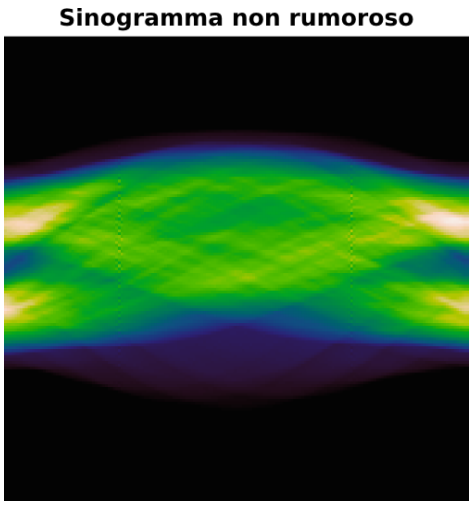
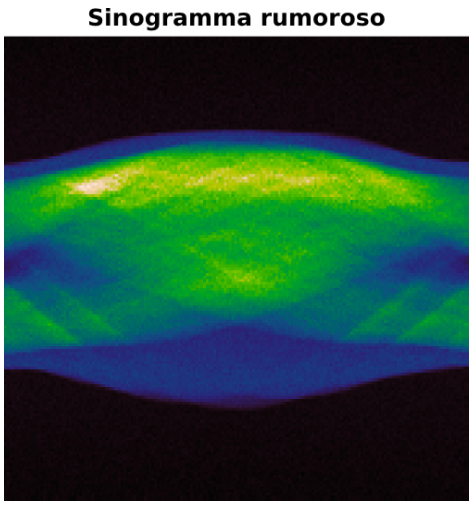


Figura 9: Fantoccio cerebrale



(a) Sinogramma ideale non rumoroso



(b) Sinogramma con artefatti e rumore Poissoniano

Tabella 1: VALUTAZIONE DELL'ACCELARAZIONE OTTENUTA CON ALGORITMO OS-EM

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

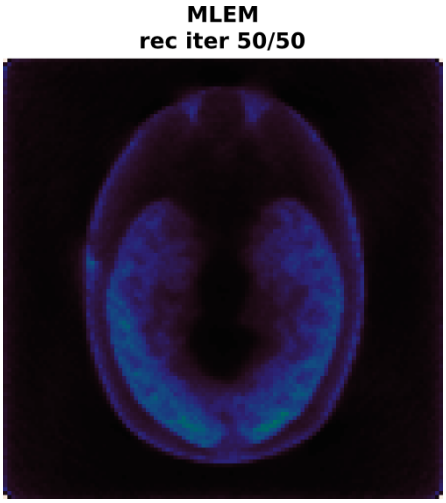


Figura 10: Ricostruzione MLEM senza correzioni

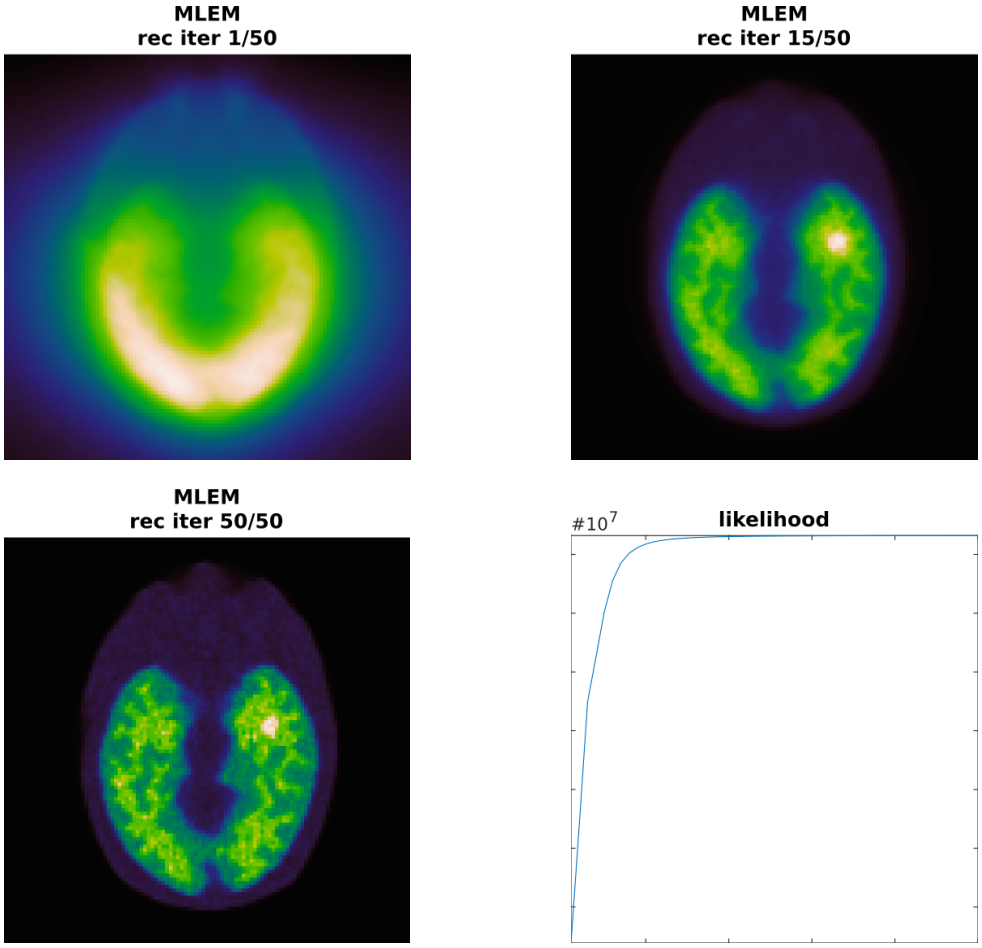


Figura 11: Ricostruzione MLEM

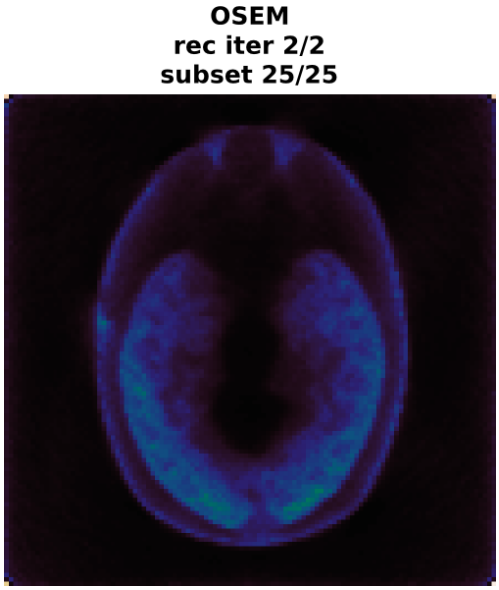


Figura 12: Ricostruzione OSEM senza correzioni

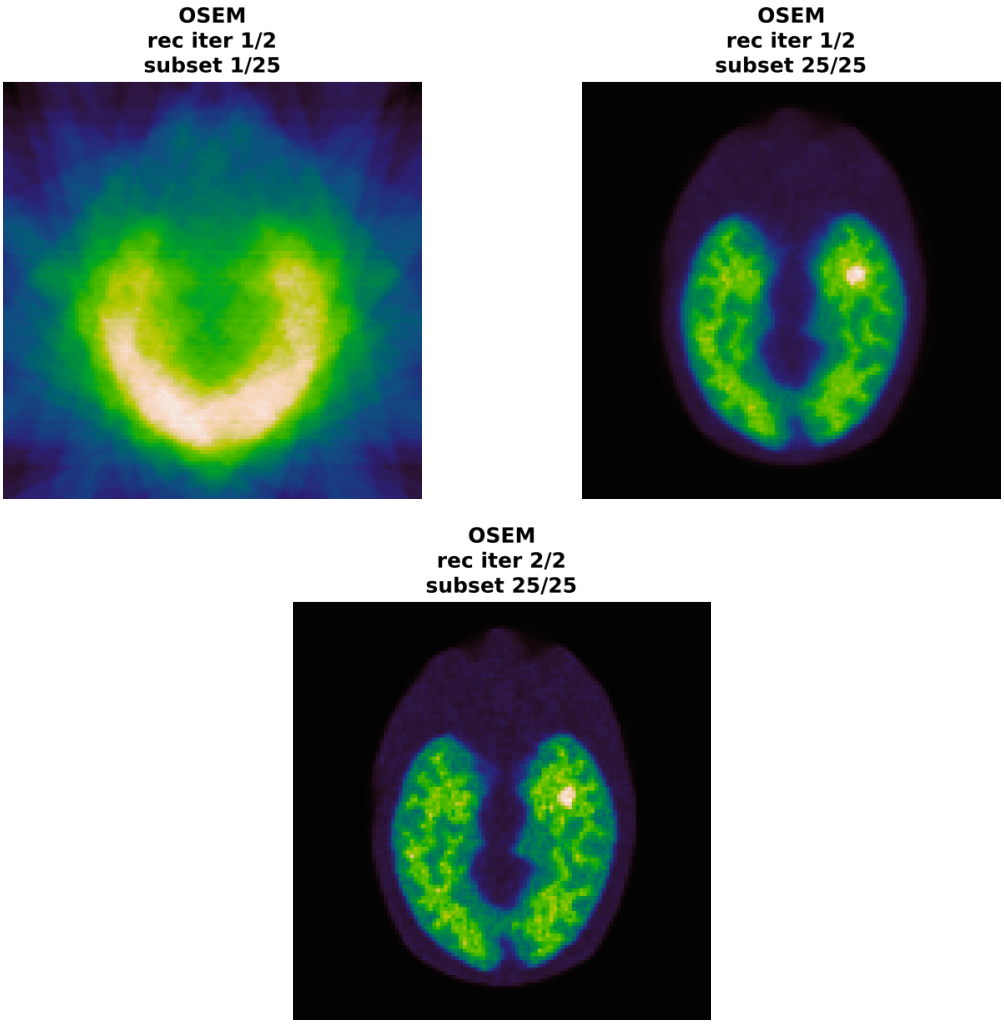


Figura 13: Ricostruzione OSEM

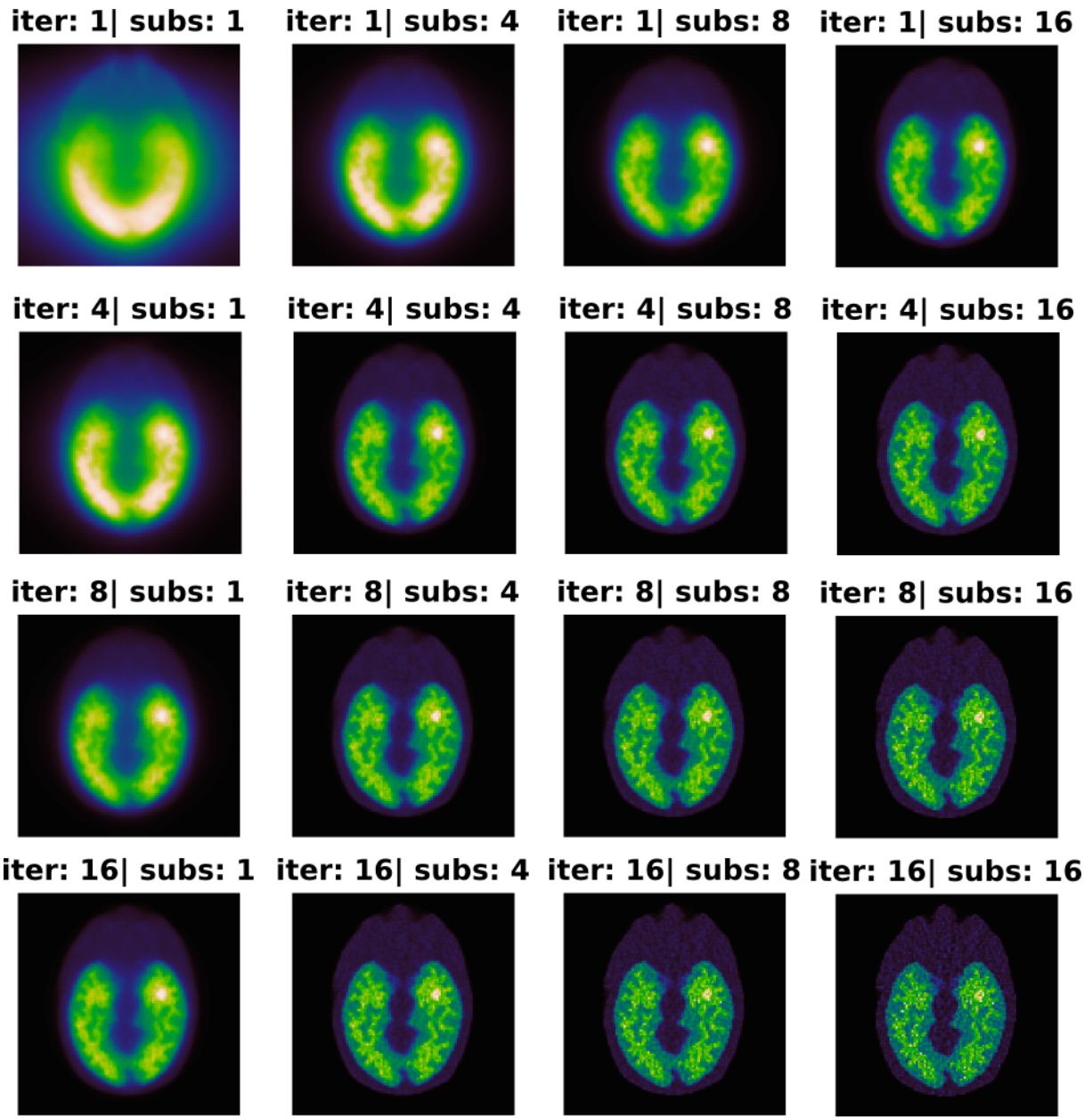


Figura 14: Relazione tra numero di subset e numero di iterazioni di algoritmo OSEM

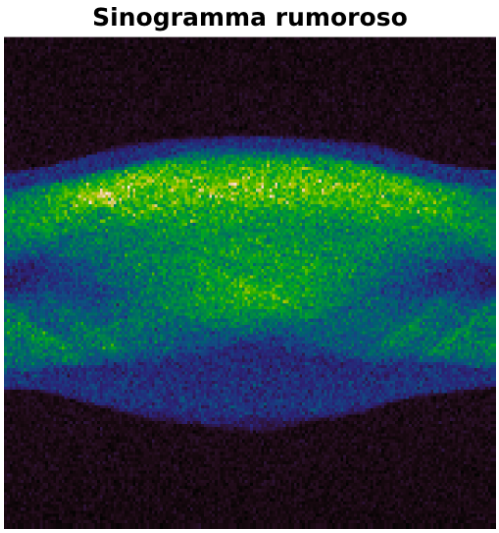


Figura 15: Riduciamo i conteggi nel sinogramma per enfatizzare l'effetto del prior

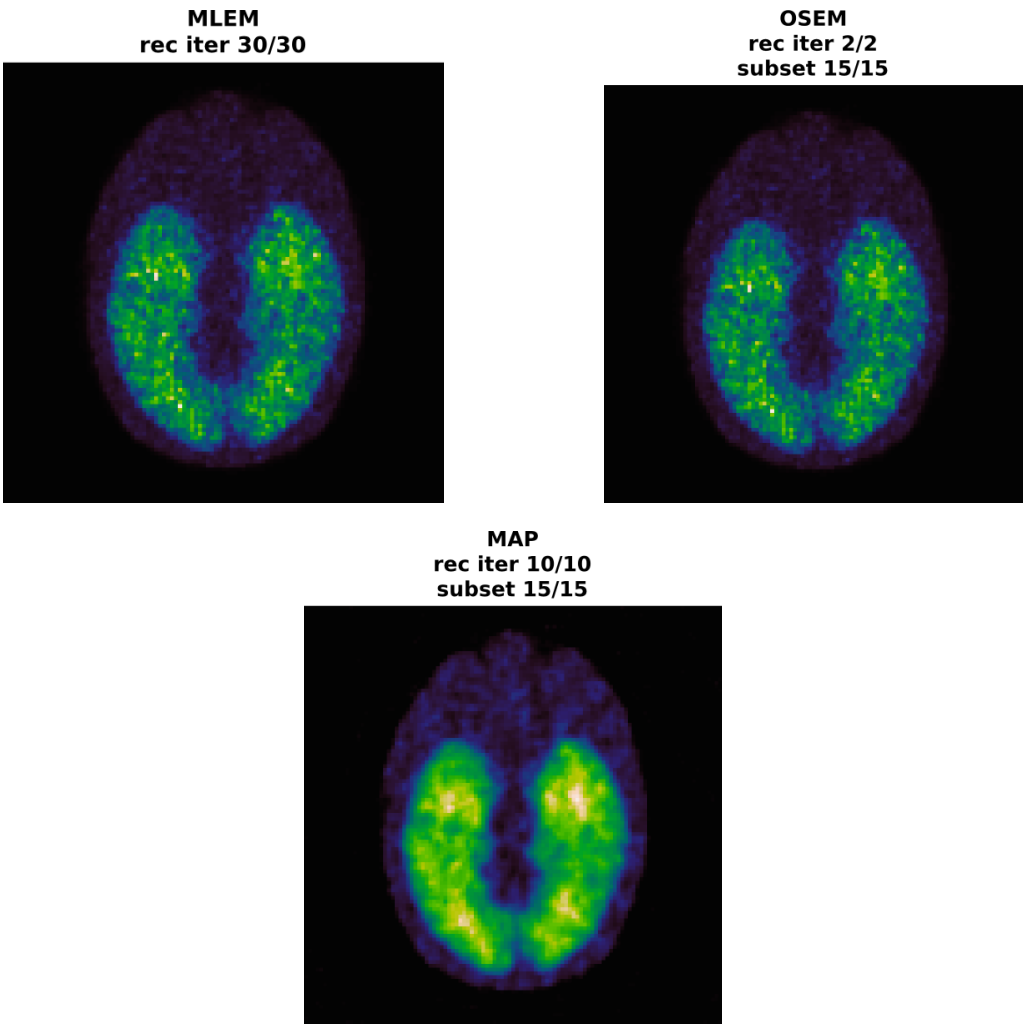


Figura 16: Ricostruzione MAP-OSL