

ESERCITAZIONE 7 27/04/2022 (Registrazione Immagini)

Artificial data

Nello sviluppo di algoritmi di elaborazione delle bioimmagini è comune l'uso di set di dati generati per via informatica, i cosiddetti **artificial data** o **synthetic data**.

Si tratta di immagini generate attraverso un software in modo da essere il più possibile simili ad immagini reali. I vantaggi dei synthetic data sono numerosi:

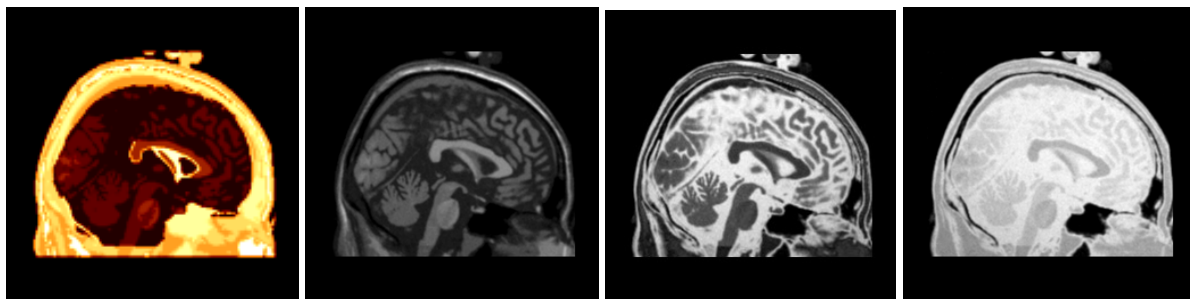
1. Non sono necessarie le autorizzazioni ed i consensi informati obbligatori per l'uso delle immagini reali sia da volontari che da pazienti.
2. Può essere generato un numero di immagini grande a piacere
3. Possono essere variati facilmente i parametri di acquisizione, come l'SNR ed il CNR.
4. Il "gold standard" per la validazione dell'algoritmo è perfettamente noto

L'ultimo punto in particolare libera il processo di validazione dall'incertezza sul "gold standard" propria della validazione fatta da utenti esperti che è sempre affetta dalla variabilità inter- ed intra-osservatore. Un modo per produrre dati sintetici è quello utilizzare il modello teorico dell'immagine biomedica. In questo caso si definirà una immagine ideale che sarà tipicamente estratta da un atlante anatomico, come il 4D Extended Cardiac-Torso (XCAT) Phantom o l'atlante di Talairach. Partendo dall'immagine ideale che costituirà il "gold standard" si aggiungono i fattori di corruzione dell'immagine ideale:

- Rumore biologico sui tessuti non isotropi
- Effetto volume parziale (filtraggio gaussiano)
- Campo di attenuazione
- Rumore

Si ottiene così una immagine realistica.

Metodi più perfezionati comportano la simulazione numerica del processo di acquisizione. In questo caso partendo dall'immagine ideale viene simulata la fisica del processo di acquisizione in dipendenza dal metodo di acquisizione utilizzato. Consideriamo ad esempio il simulatore MR cerebrale brainweb (<https://brainweb.bic.mni.mcgill.ca/>).



Atlante

T1

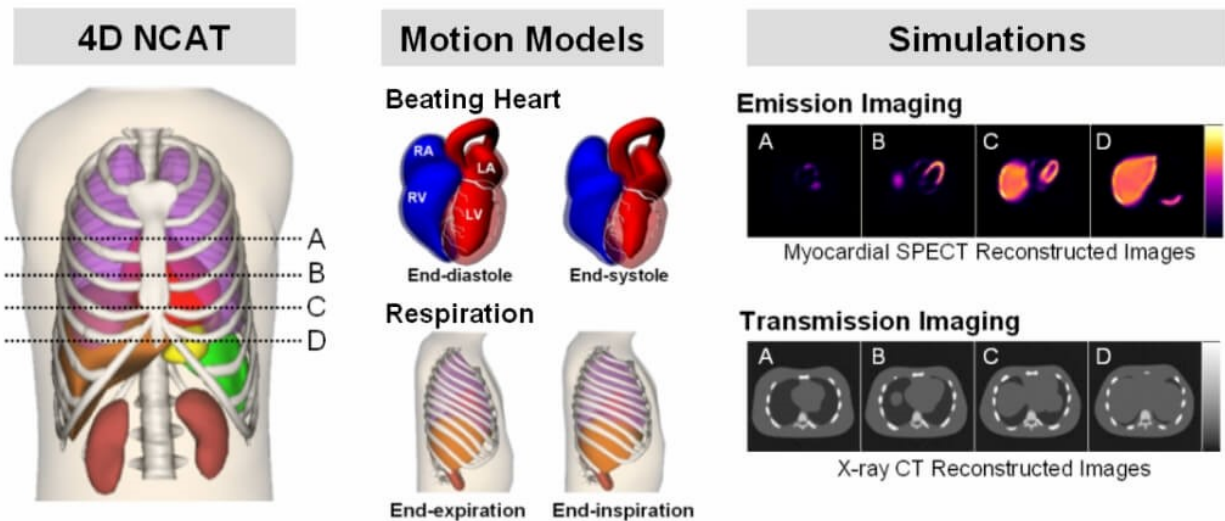
T2

PD

Il simulatore parte da un atlante anatomico (a) e permette di definire il tipo di sequenza (T1, T2, PD), il thickness di fetta, il rumore, e la disomogeneità del campo RF.

Esistono altri simulatori MR free, come JEMRIS (<http://www.jemris.org/>) che fornisce anche una interfaccia MATLAB, coreMRI (<https://www.coremri.org/>) ed altri.

Il 4D-MCAT phantom simula una acquisizione SPECT
<https://www.hopkinsmedicine.org/radiology/research/divisions/radiological-physics/research/projects/imaging-simulation-computer-phantoms.html>



CTsim implementa la simulazione di immagini CT (<http://www.ctsim.org/>)

Sono disponibili anche immagini US simulate: <https://team.inria.fr/epione/en/data/stras/>

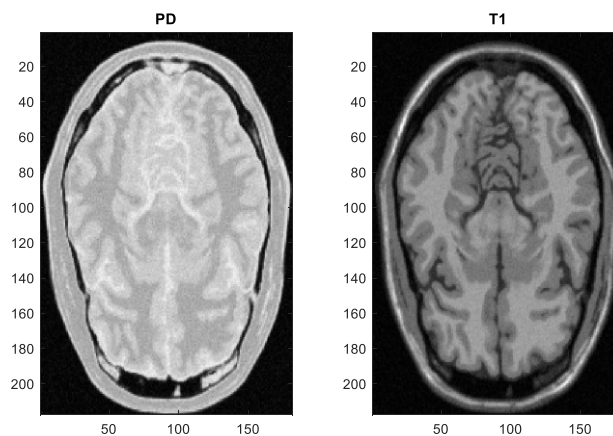
La disponibilità di synthetic data permette di fare una prima valutazione del funzionamento di un algoritmo di elaborazione variando le caratteristiche delle immagini in modo efficiente ed economico.

Lo scopo dell'esercitazione è sviluppare una procedura di registrazione basata sugli algoritmi genetici che esegua la registrazione 2D di immagini MR. Le immagini da registrare sono quelle di un cosiddetto "synthetic phantom", cioè un phantom generato via software. In particolare utilizziamo immagini dal sito **brainweb** prima descritto:

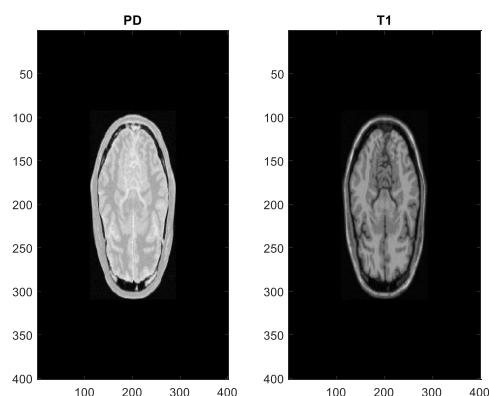
- `t1_icbm_normal_1mm_pn3_rf0.mnc` (T1 pesata)
- `pd_icbm_normal_1mm_pn3_rf0.mnc` (PD)

Le immagini sono in formato MINC (McConnell Brain Imaging Center (McBIC) at the Montreal Neurological Institute (MNI)), un formato proprietario del centro che ha realizzato il simulatore brainweb. Possono essere importate in MATLAB con la funzione **loadminc** ottenibile dal sito MATLAB exchange (<https://it.mathworks.com/matlabcentral/fileexchange/32644-loadminc>).

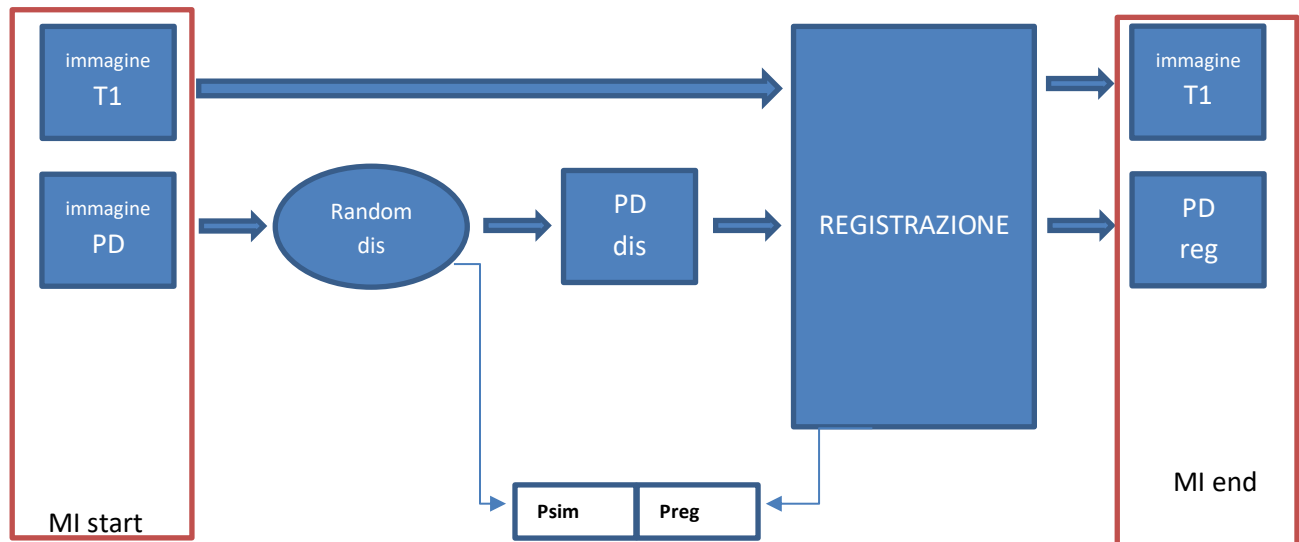
Per ridurre i tempi di calcolo consideriamo una singola slice. Quindi andremo ad estrarre dai due volumi una slice centrale, ad esempio la slice 62 ottenendo due immagini 2D T1 e PD.



Come si osserva BrainWeb genera immagini con un FOV corrispondente all'area cerebrale. Visto che dobbiamo simulare un disallineamento è opportuno effettuare uno zero padding che consenta di spostare le immagini senza perderne una parte (**padarray**) e renda la matrice quadrata.



Partendo da questa coppia di immagini, vogliamo realizzare un simulatore che simuli un disallineamento rigido noto (roto-traslazione in 2D), esegua il riallineamento delle due immagini e misuri la differenza tra i parametri simulati e quelli trovati dall'algoritmo di registrazione. La flow chart dell'algoritmo risulta:



All'inizio abbiamo le due immagini che saranno caratterizzate da un certa simiglianza, ad esempio il valore della mutua informazione MI start. Una immagine (la PD nell'esempio) funge da immagine floating e viene sottoposta ad una rototraslazione casuale di cui vengono memorizzati i tre parametri Psim. L'immagine floating viene registrata con l'immagine T1 da un algoritmo di registrazione che produrrà l'immagine floating registrata ed i parametri di correzione del disallineamento Preg. Inoltre potremo calcolare la MI tra le immagini registrate (MI end).

Se la registrazione è corretta, dovremmo trovare $Psim = - Preg$ ed $MIstart = MIend$.

La bontà dell'algoritmo di registrazione può quindi essere valutata dalle differenze $(MI\ start - MI\ end)$ e $(Psim - Preg)$.

Iterando il procedimento ad ogni iterazione avremo un disallineamento diverso e quindi potremo verificare l'algoritmo in situazioni diverse facendo un'analisi di tipo statistico.

Una volta messo a punto l'algoritmo, per valutarne le prestazioni possiamo quindi eseguirlo per un numero N di volte (ad esempio $N=20$) ed annotare i parametri imposti e quelli calcolati per le sei incognite rappresentate dai parametri di rototraslazione. I dati raccolti possono essere analizzati tracciando i relativi sei Bland-Altman plot. Esistono varie funzioni che implementano il Bland Altman plot su File Exchange, ad esempio https://www.mathworks.com/matlabcentral/fileexchange/71052-blandaltmanplot?s_tid=srchtitle

Per quanto riguarda la MI il valore MIstart sarà sempre lo stesso, per cui potremo valutare media ed SD della differenza tra MIend e MIstart, che teoricamente dovrebbe essere nulla.

La procedura di disallineamento random è operata dalla funzione **disImage2D** che prende in ingresso una immagine e restituisce l'immagine roto-traslata ed i parametri di disallineamento.

La procedura di registrazione al solito è caratterizzata da tre parametri:

- 1) Search space: le due triplette di valori che definiscono le due roto-traslazioni (6 parametri). La massima traslazione indotta da **disImage2D** è +/- dim/10, dove dim è la dimensione dell'immagine, mentre il massimo range di rotazione è [-60°, +60°].
- 2) Metrica: essendo variabile la distribuzione dei livelli di grigio sarà opportuno usare la mutua informazione. Avendo due immagini e prendendo la prima immagine I1 come riferimento la metrica per l'allineamento sarà MI(I1,I2).
- 3) Ottimizzatore: Algoritmo genetico (minimizzatore)

L'algoritmo dovrà stimare correttamente i parametri di trasformazione $P=[tx1 \ ty1 \ a1]$ che allineano correttamente l'immagine I2 su I1.

L'algoritmo GA (Genetic Algorithm) è implementato in MATLAB nel toolbox **Global Optimization**.

L'algoritmo GA è infatti un ottimizzatore globale. La funzione MATLAB è:

```
x = ga(fun,nvars,A,b,Aeq,beq,lb,ub,nonlcon,options)
```

dove:

1. **fun** è la funzione da ottimizzare. È necessario definire una funzione MATLAB (**function**) che prenda in ingresso i parametri della funzione e restituisca in uscita il valore. Nel caso di un problema di registrazione, i parametri saranno i parametri per la registrazione ed in uscita avremo il valore della metrica da valutare, ad esempio la mutua informazione.
2. **nvars** è il numero di parametri della funzione, ad esempio 3 nel caso di trasformazione rigida 2D
3. **A e b** definiscono una disuguaglianza lineare, per i nostri scopi si pone $A=[]$, $b=[]$
4. **Aeq e Beq** definiscono una disuguaglianza non lineare, per i nostri scopi si pone $Aeq=[]$, $beq=[]$
5. **lb e ub** definiscono i limiti dei parametri di registrazione, quindi il search space. Il setting di questi parametri è importante per il funzionamento dell'algoritmo.
6. **nonlcon** definisce un indice per le disuguaglianze lineari o non lineari, poniamo $nonlcon=[]$
7. **options** è un oggetto creato da **optimoptions** che definisce i parametri dell'algoritmo genetico (`optimoptions('ga')`).

La struttura options contiene i parametri per il setting dell'algoritmo genetico, quali:

- EliteCount (numerosità dell'elite),
- CrossoverFcn (funzione usata per il crossover)
- CrossoverFraction (percentuale di individui prodotti via crossover),
- MaxGenerations (numero massimo di iterazioni),
- MaxStallGenerations (numero massimo di generazioni senza cambio della migliore fitness),
- MutationFcn (tipo di mutazione),
- PopulationSize (numero di individui)
- UseParallel (parallelizza l'algoritmo se il computer lo supporta)

- UseVectorized (velocizza l'algoritmo se la fitness può essere calcolata su tutta la popolazione allo stesso tempo)
- Display (permette di visualizzare i risultati iterazione per iterazione)
- 'PlotFcn' (permette di graficare la fitness durante le iterazioni)

Come detto, data la natura delle immagini, è opportuno usare la Mutua Informazione (MI) come metrica di registrazione, ricordando che dovremo cercare il massimo della MI e non il minimo, quindi una volta realizzata la funzione che computa la MI tra due immagini (ad esempio calcolaMI) la fitness function nell'algoritmo GA verrà definita come:

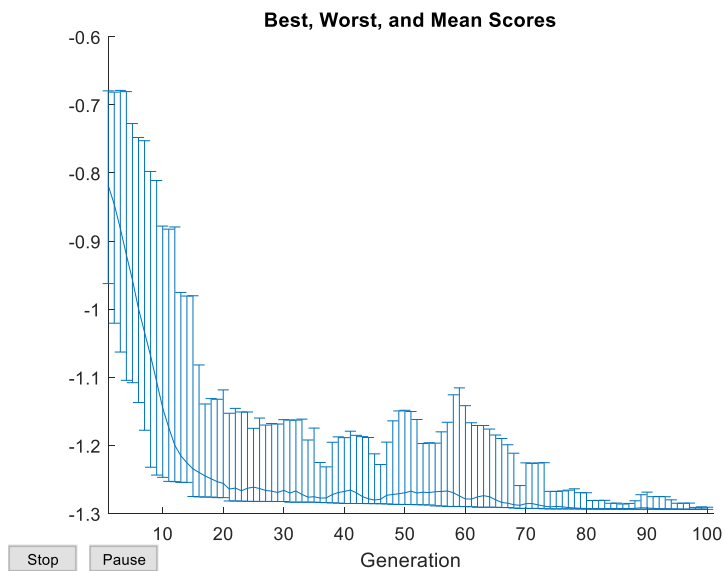
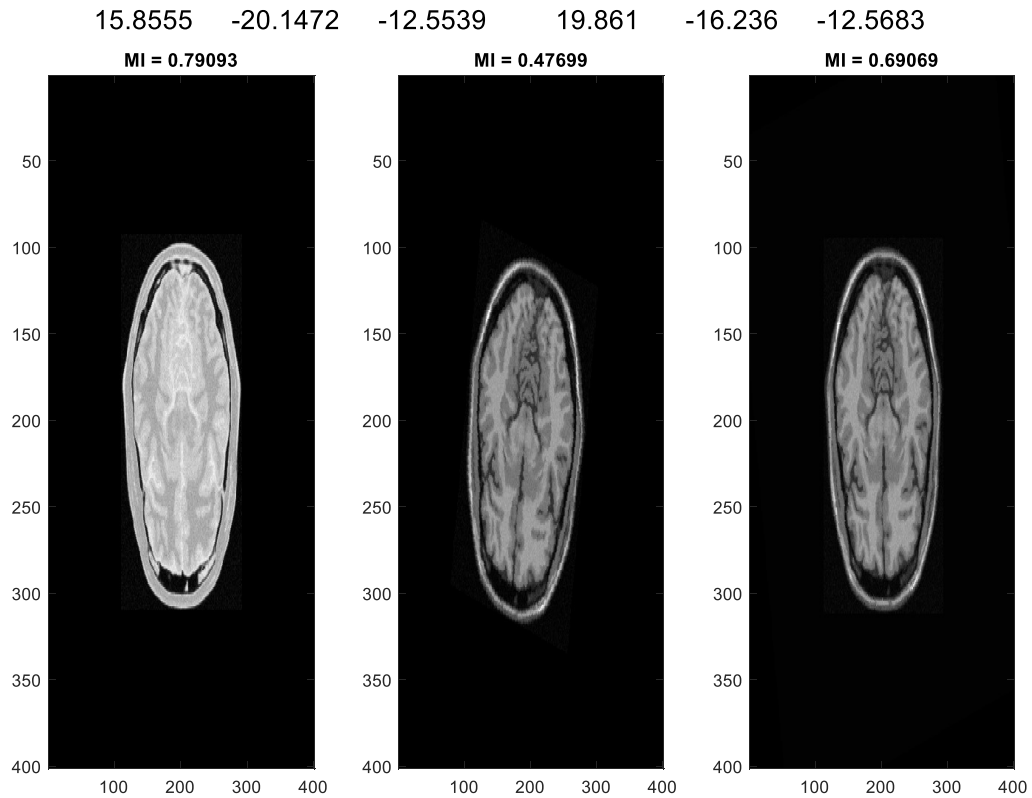
```
FitnessFunction = @(par) -calcolaMI(par,I1,I2);
```

dove par è un vettore a tre elementi che rappresenta i parametri di rototraslazione.

L'algoritmo GA troverà i parametri par (la trasformazione tra le immagini, 3 elementi) sulla base di I1 e I2 che sono le immagini da registrare. Il meno davanti alla funzione esegue la massimizzazione invece che la minimizzazione. Formulare la FitnessFunction in questo modo indica all'algoritmo quali sono i parametri da ottimizzare (par) e quelli fissi (le due immagini).

La funzione calcolaMI dovrà roto-traslare l'immagine I2 e calcolare la MI tra I1 e I2 rototraslata. Come al solito nel calcolo della mutua informazione bisogna limitare il calcolo alle parti comuni delle immagini e utilizzare un algoritmo di interpolazione di tipo NN.

Un esempio di run dell'algoritmo da implementare è il seguente, chiaramente visto l'utilizzo di un disallineamento random ogni run sarà differente da un altro.



Ottimizzazione

In questo run abbiamo una buona correzione della rotazione mentre un errore non trascurabile per le traslazioni. La MI iniziale era 0.79, quella dopo il disallineamento 0.48 e quella finale 0.69.