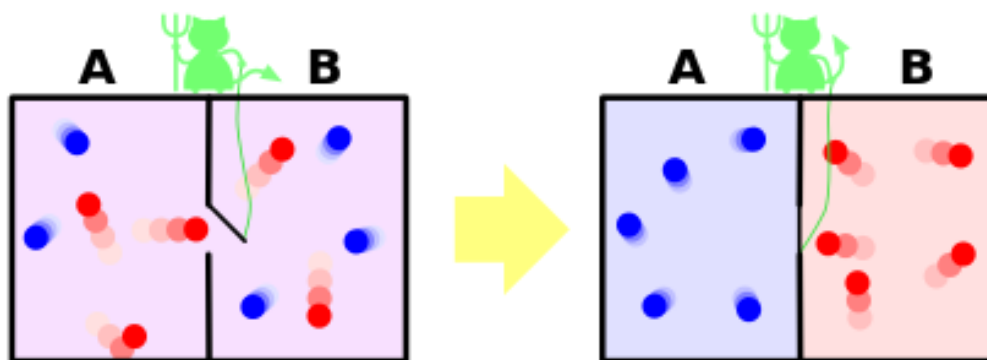


ESERCITAZIONE 8 04/05/2022 (Registrazione Serie Temporal)

Daemons Algorithm

Introduciamo l'algoritmo di registrazione implementato nella funzione Matlab *imregdemons*, che utilizzeremo nell'esercitazione. L'algoritmo implementa una registrazione locale di tipo warping attraverso l'algoritmo "daemons", ispirato ai "diavoletti di Maxwell" della teoria dei gas (Thirion, JP, Image matching as a diffusion process: an analogy with maxwell's demons. Medical Image Analysis 2(3) (1998)). L'algoritmo esegue la registrazione attraverso l'estrazione di features corrispondenti ad alti valori del gradiente, per cui è di tipo topologico.



Nell'esperimento ideale di Maxwell si considerano due contenitori immaginari, A e B, riempiti con un gas identico e alle stesse temperature, posti uno a fianco dell'altro, separati solamente da una piccola botola apribile con cui è possibile mettere in comunicazione i due vani. Un piccolo diavoletto sta a guardia della botola, mantenendola chiusa e osservando le molecole agitarsi nei due diversi contenitori. Quando una molecola più veloce delle altre si dirige verso la botola, il diavoletto la apre e lascia che la molecola passi dal contenitore A al contenitore B. La velocità media delle molecole in B ne risulta quindi ogni volta aumentata, mentre quella delle molecole in A ne esce diminuita. Come noto, all'aumento della velocità media delle molecole, corrisponde un aumento della temperatura. Pertanto, successivi interventi del diavoletto comportano la diminuzione della temperatura in A, e l'aumento di quella in B: questo è in contraddizione con la seconda legge della termodinamica. L'idea è basata sul fatto che il secondo principio, a differenza del primo, ha carattere statistico. Se si accetta di poter descrivere un gas (o in generale un corpo macroscopico) come un insieme di particelle (eventualmente interagenti) si può reinterpretare lo stato di equilibrio termodinamico di un sistema chiuso come quello più probabile e quindi quello più di frequente realizzato dalle particelle. Nulla vieta l'esistenza di fluttuazioni termodinamiche che possono portare il sistema in uno stato diverso da quello di equilibrio: esse sono escluse solo sulla base della loro improbabilità, non per ragioni fisiche codificate nelle leggi della meccanica. Il diavoletto dovrebbe allora essere un congegno di qualche tipo, operante secondo tali leggi, ma a livello microscopico. L'esperimento ideale è vanificato dal fatto che la conoscenza della velocità delle particelle richiede una misura che comporta un consumo di energia e quindi un aumento globale di Entropia del sistema. (Wikipedia).

Da un punto di vista algoritmico, siano R l'immagine reference e F l'immagine floating. L'idea dell'algoritmo è che un pixel di F si muove nel tempo cercando di minimizzare la differenza tra R e F .

L'algoritmo procede nel modo seguente in modo iterativo:

- Si pone il passo di iterazione $n=1$. Si pone $F(1)=F$
- Si calcola lo spostamento al passo n come:

$$U(n) = \frac{(F(n) - R)\nabla R}{\|\nabla R\|^2 + \alpha(F(n) - R)^2}$$

Quindi lo spostamento è diverso da zero se le due immagini al passo n sono diverse e se siamo in una regione dell'immagine R con gradiente ∇R non nullo. Lo spostamento è tanto maggiore quanto maggiore è il gradiente e la differenza tra le immagini. Il parametro α non nullo serve a migliorare la convergenza dell'algoritmo contrastando la divergenza di $U(n)$ per gradiente di R nullo.

- Si filtra $U(n)$ con un filtro di smoothing (gaussiano) ottenendo $\bar{U}(n)$.
- Si calcola $F(n+1)$ applicando $\bar{U}(n)$ a $F(n)$
- Si pone $n=n+1$ e si ricomincia dal passo 2.

L'algoritmo è multi-scala, quindi inizia ad operare su di una versione a bassa risoluzione dell'immagine e poi raffina la registrazione incrementando la risoluzione.

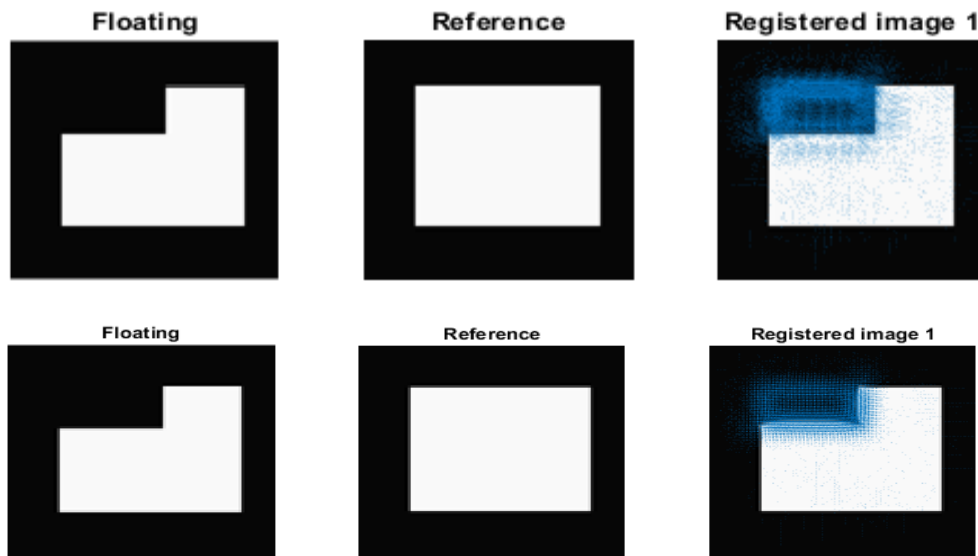
La sintassi della funzione è:

[D, registered] = imregdemons(floating,reference,[N])

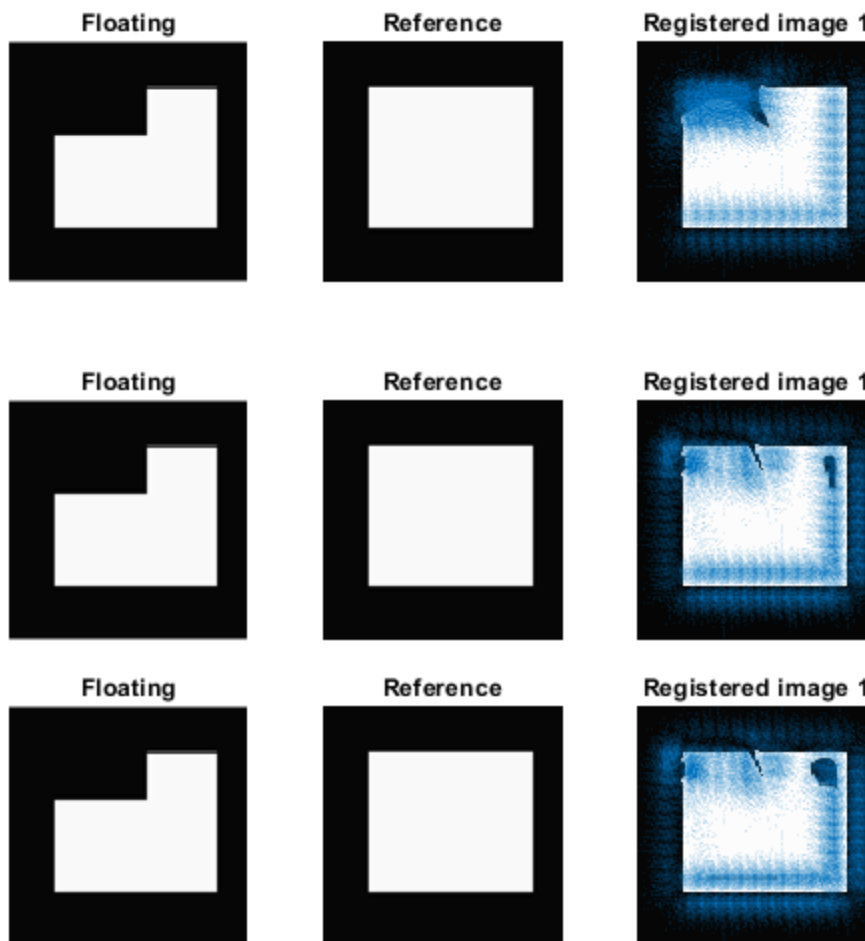
La funzione prende come input una coppia di immagini (reference e floating) e restituisce l'immagine floating registrata (registered) e il "displacement field" D , che contiene la traslazione imposta a ciascun pixel dell'immagine. Nel caso di registrazione 2D il campo D avrà due componenti corrispondenti alle traslazioni lungo x e y .

Come detto l'algoritmo usa un approccio "piramidale", cioè inizia a lavorare su versioni a risoluzione ridotta delle immagini trovando un valore di D approssimato e poi lo raffina incrementando la risoluzione fino ad arrivare all'immagine con la risoluzione originale. Il numero di livelli è definito dal parametro `PyramidLevels` (default 3). Il numero di iterazioni è definito dal parametro opzionale N (default 100). Essendo l'algoritmo multi-scala N può essere un vettore contenente il numero di iterazioni per ciascuna scala. Il parametro 'AccumulatedFieldSmoothing' determina la deviazione standard del filtro di smoothing, che determina il livello di dettaglio del processo di deformazione dell'immagine.

Consideriamo le due immagini in figura, al primo passo il campo di spostamento è concentrato sulla regione diversa tra le due immagini, ed in particolare su bordo della reference.



L'immagine floating comincia a deformarsi sulla base del campo di spostamento.



La procedura continua fino alla registrazione

Guida all'esercitazione

Scopo dell'esercitazione è realizzare la registrazione di una serie temporale di immagini biomediche. In questo caso la serie temporale è rappresentata da immagini MR che descrivono la perfusione renale con mezzo di contrasto. Nella directory `RENAL_PERF` troviamo quindi una serie 2D+T contenente immagini 2D acquisite a 70 frame temporali diversi.

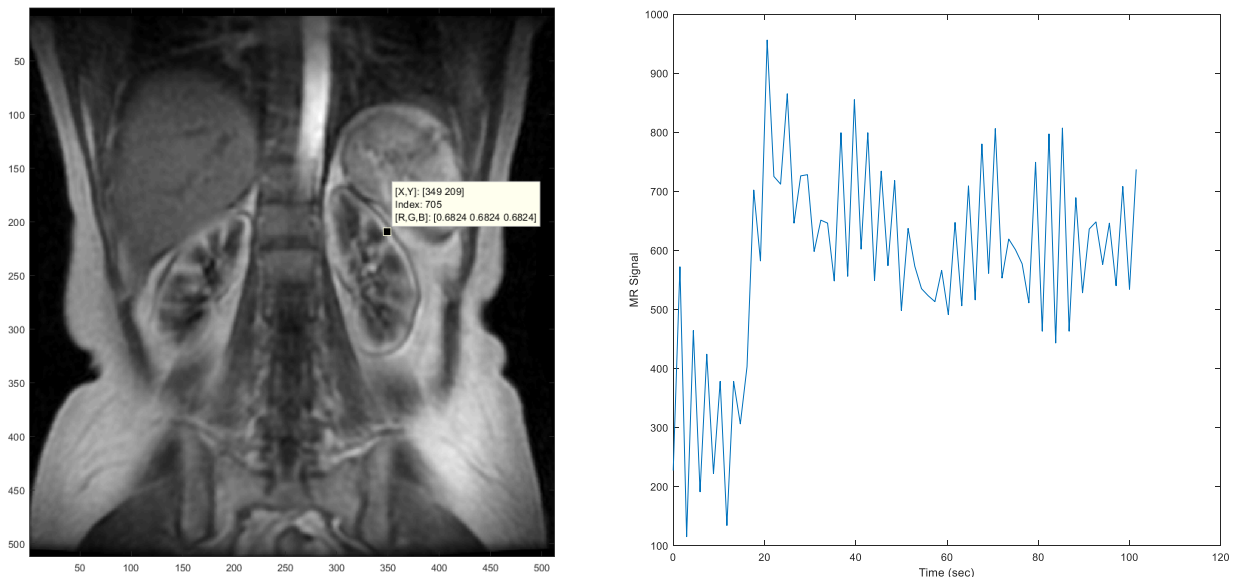


La serie è parte di una acquisizione completa 3D+T a più fette parallele che coprono la regione renale, il tempo di campionamento è quello necessario per acquisire tutte le fette parallele. Il tempo di acquisizione dei frame temporali viene scritto nel campo DICOM `TriggerTime`, espresso al solito in millisecondi. Come si osserva, l'intervallo di campionamento temporale è dell'ordine del secondo e mezzo.

Come si osserva dalle immagini, la posizione dei reni varia da frame a frame seguendo il movimento respiratorio del paziente sotto esame che è in respiro libero. La sequenza di immagini può essere visualizzata con la funzione MATLAB *sliceViewer*, o *implay* normalizzando il

volume nel range 0:255 e settando l'opzione colormap->"Specify range of displayed pixel values".

Il disallineamento delle immagini costituisce un problema dal punto di vista diagnostico, ad esempio se è necessario estrarre una curva di perfusione da una particolare zona anatomica.



In figura è illustrata la curva di perfusione di un pixel della corteccia renale che rappresenta la regione di interesse. Come si osserva la curva presenta importanti artefatti dovuti al movimento del rene rispetto al riferimento dell'immagine.

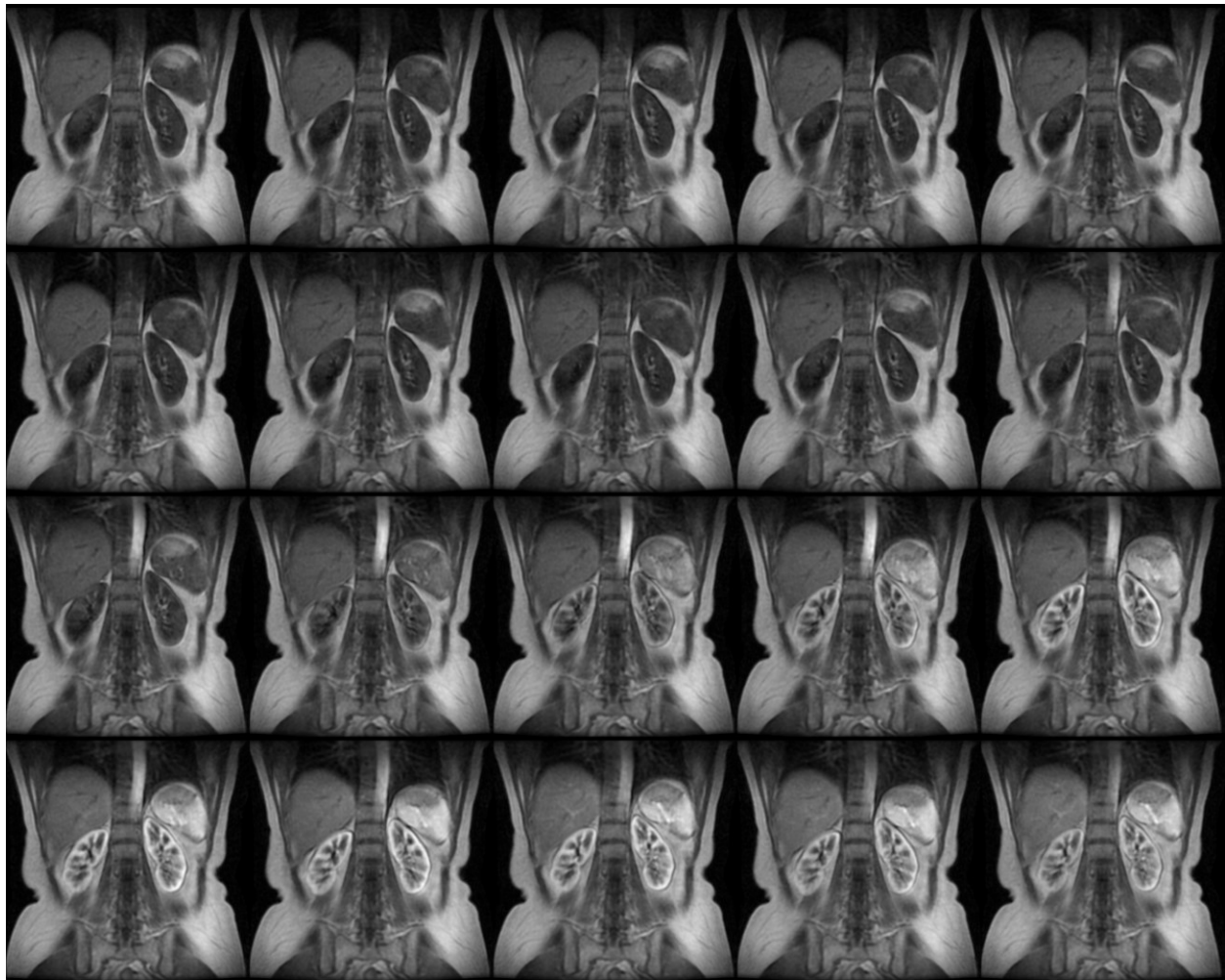
Per estrarre correttamente la curva di perfusione abbiamo quindi bisogno di implementare un algoritmo di registrazione di una serie temporale, che come sappiamo viene ricondotta ad una serie di registrazioni tra coppie di immagini. Iniziamo ad esaminare quindi le tre componenti della registrazione tra due immagini della serie:

- Spazio di Ricerca: Come si osserva delle immagini, la deformazione indotta dal movimento respiratorio è tipicamente locale (o non rigida). Infatti, alcune strutture come i reni si muovono mentre altre come la spina dorsale ed il grasso sub-cutaneo sono sostanzialmente ferme. Quindi è necessario utilizzare un algoritmo di registrazione locale.
- Metrica: Le immagini sono evidentemente unimodali. Tuttavia, essendo presente un mezzo di contrasto che altera i livelli di grigio sarebbe opportuno utilizzare una metrica che sia legata alla topologia dei tessuti invece che al loro segnale, come ad esempio la mutua informazione. Tuttavia l'uso della MI in una registrazione locale può essere troppo pesante dal punto di vista computazionale.
- Ottimizzazione: Il disallineamento delle immagini è limitato quindi è ragionevole utilizzare un ottimizzatore locale.

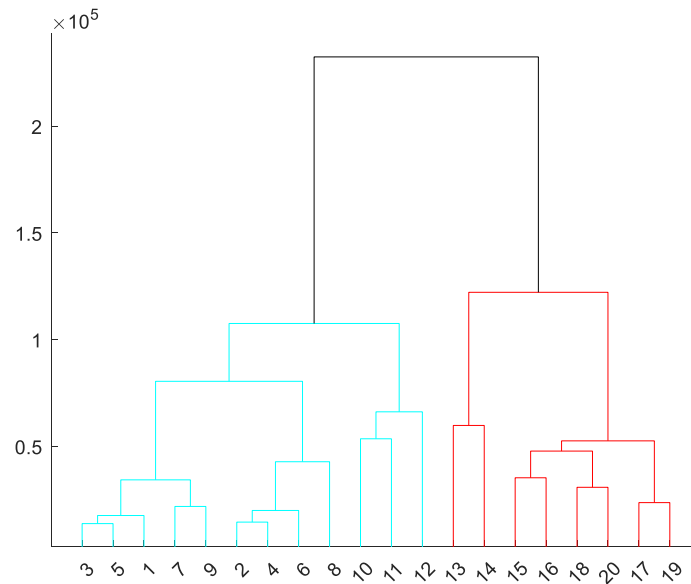
Appare quindi ragionevole l'uso della funzione *imregdemons*, che esegue la registrazione attraverso l'estrazione di features corrispondenti ad alti valori del gradiente, per cui è di tipo topologico. Il problema della funzione *imregdemons* è che può essere ingannata dalla differenza di livelli di grigio indotta dal contrasto.

Come si osserva dalla nostra sequenza di immagini l'intervallo di acquisizione (circa 1 sec) è dello stesso ordine di grandezza del periodo del fenomeno che induce il disallineamento (il respiro del paziente), per cui la differenza tra due frame temporali consecutivi è rilevante e l'approccio di registrare ogni immagine con la precedente, minimizzando la differenza dovuta al contrasto, non è ottimale.

Possiamo invece procedere classificando le immagini con un algoritmo di clustering gerarchico, che crei un albero che descrive la simiglianza delle immagini, e poi procedere alla registrazione seguendo l'albero. Per semplicità nell'esercitazione sono considerate le prime 20 immagini della sequenza, dove si riconoscono immagini "basali" (con poco o nullo contrasto) e immagini dove il contrasto è ben visibile.



Per ottenere il clustering gerarchico dobbiamo definire una matrice delle distanze $D_{N \times N}$ (N numero di frame temporali), dove $D(i,j)$ è la distanza tra le immagini i e j . Visto che vogliamo individuare due classi di immagini (con e senza contrasto) conviene usare come metrica l'errore quadratico medio, in modo da esaltare la differenza tra le due classi. Calcolata D , la funzione MATLAB *linkage* crea l'albero gerarchico T che può essere visualizzato con *dendogram*. L'aggregazione dei cluster viene governata dall'opzione 'method' di linkage ('single', 'complete', 'average').



Dovremmo ottenere un albero del tipo in figura (qui è stata usata “complete” come opzione), dove l'algoritmo di clustering ha diviso le immagini in due gruppi pre- e post- contrasto. La divisione tra gruppi si ottiene con la funzione *cluster*.

Se andiamo ad esaminare l'albero, notiamo come il clustering gerarchico abbia diviso le immagini in due gruppi (1-12 e 13-20) identificando i livelli di contrasto, ma abbia anche diviso le immagini pari da quelle dispari individuando il movimento dei reni.

A questo punto possiamo registrare tra loro le immagini dei due cluster che hanno livelli di grigio simili, prendendo una immagine come riferimento e registrando le altre rispetto ad essa. Otterremo due gruppi di immagini registrate tra loro. Per ottenere una buona registrazione è necessario settare correttamente i parametri dell'algoritmo *imregdemons*. In particolare, dobbiamo evitare di avere immagini registrate non realistiche, come avviene per un livello di smoothing troppo piccolo. Avremo quindi come risultato due serie di immagini (senza contrasto e con contrasto) registrate sulla singola serie ma non tra loro.

Per registrare le sue serie dobbiamo operare secondo il principio del clustering gerarchico, scegliendo due immagini rappresentative delle due serie. Se abbiamo usato un approccio “complete” saranno le due immagini più “lontane”, se abbiamo scelto un approccio “single” quelle più vicine etc. Quindi calcoliamo tutte le distanze incrociate tra le immagini delle due

serie registrate, identifichiamo le due “rappresentanti” e facciamo la registrazione con **imregdemons**, scegliendo in modo arbitrario l’immagine di riferimento e quella floating. A questo punto applichiamo il “displacement field” ottenuto dalla registrazione con **imwarp** a tutte le immagini della serie che conteneva l’immagine floating ed otteniamo la serie finale registrata. Estruendo una curva di perfusione corticale dalla serie registrata dovremmo ottenere una riduzione visibile delle oscillazioni del segnale (in figura è mostrato il pixel (220,356)).

