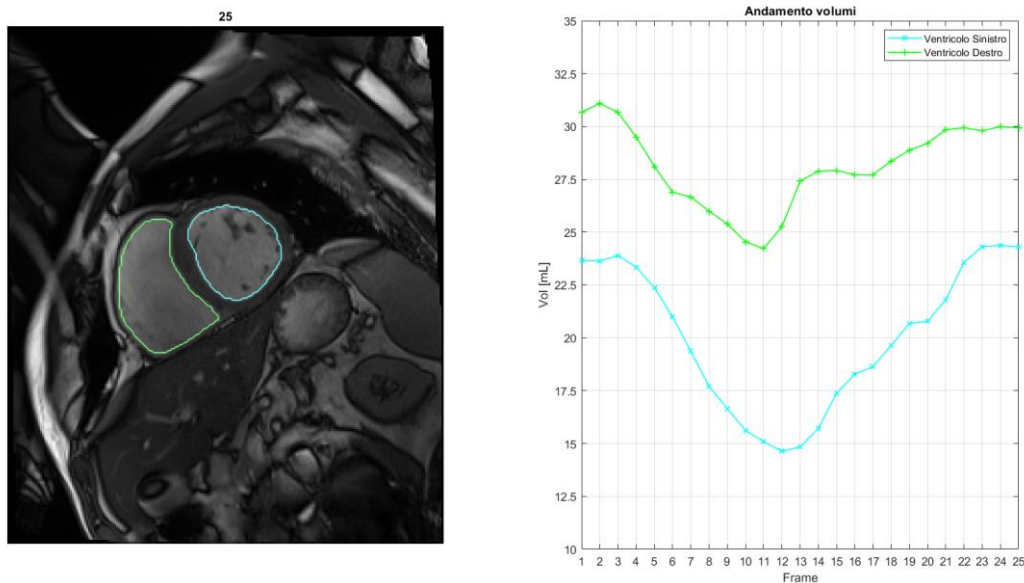


Esercizio 3

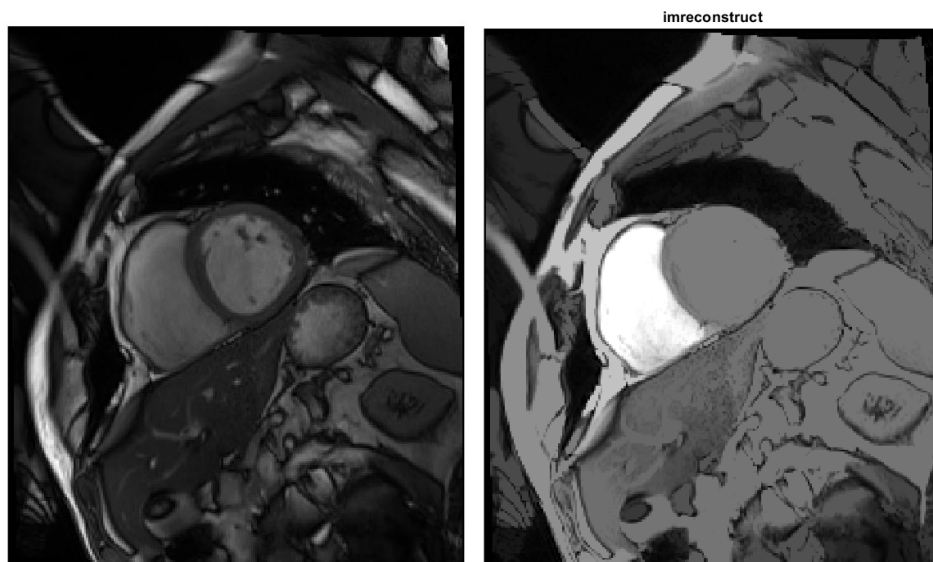
Lo script ottiene la segmentazione di ventricolo destro e sinistro, mostrandoli prima separatamente e poi insieme, con un aggiornamento visivo per simulare un video (25Hz); viene aggiornata sia la presentazione della segmentazione, che il volume rappresentato sul plot. La presentazione combinata viene ripetuta una ulteriore volta per dare la possibilità di valutare meglio i particolari. Eventualmente si può facilmente rimuovere il tempo dalla pausa (*pause()*) per poter procedere frame per frame manualmente. Alla fine viene proposto anche un andamento fittato per entrambe le curve, ritenendo che la presenza di piccoli glitch possa causare inesattezze, le quali in considerazione di un utilizzo pratico reale andrebbero evitate.



Viene ora proposta una descrizione dell'algoritmo utilizzato. Le immagini rappresentano i risultati degli step intermedi principali eseguiti sulla *slice 21*. La 21 infatti, è stata *una* di quelle che ha richiesto più lavoro per ottenere bordi correttamente definiti.

Per realizzare la segmentazione si è proceduto tramite forte tuning per ottenere un algoritmo che riuscisse a dare buoni risultati su ogni singolo frame. Dato che era richiesto più automatismo possibile, il primo passo che è di ricostruzione morfologica (tramite *imreconstruct*) viene inizializzato da un punto scelto al centro dei ventricoli. Questo primo passaggio serve per mettere più in risalto il ventricolo, infatti ne aumenta il contrasto rispetto il resto dell'immagine e riduce le piccole oscillazioni di videointensità.

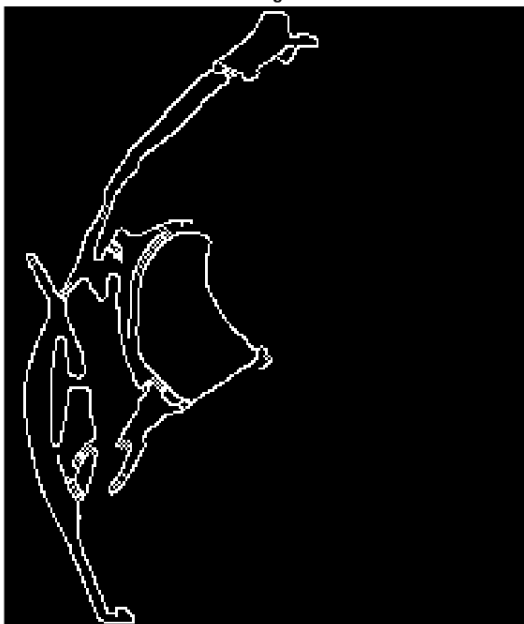
Di seguito l'immagine iniziale e quella trasformata:



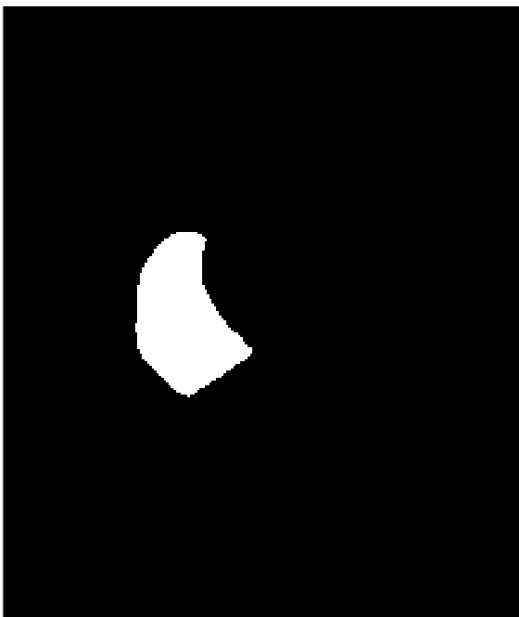
Lo step successivo scelto è stato quello di *imextendedmax*, per ottenere una maschera di pixel di massimi locali. Questa maschera ottenuta viene usata per "eliminare" (dalla slice già esaltata) parti non utili, facilitando il processo di segmentazione. L'immagine di seguito mostra la parte rimanente.



bridge



select



Il successivo step serve ancora una volta per esaltare il ventricolo; in questo caso andiamo ad applicare una trasformazione all'immagine (*imadjust*) per rimappare i livelli di intensità, andando a "tagliare" livelli di nero: esaltando così ancora una volta il ventricolo rispetto al resto e "forzando" un aumento di contrasto. A titolo di esempio, senza questo step, la successiva applicazione di *canny* avrebbe mal identificato l'angolo in basso a destra del ventricolo, credendolo parte dello stesso. Lo stesso triangolo si sarebbe potuto evitare anche con una *imreconstruct* più ristretta, causando però saltuariamente la perdita di parti reali del ventricolo nei vari frame.

Viene dunque utilizzato il metodo *canny* per trovare i contorni. Per avere una sicurezza in più che i contorni del ventricolo siano chiusi, viene applicata l'operazione morfologica di *bridge* (tramite *bwmorph*).

Nelle prime implementazioni dell'algoritmo, per assicurarsi che i bordi fossero chiusi, era necessario venire a maggiori compromessi: essere molto meno restrittivi con *canny*, oppure utilizzare operazioni morfologiche come *imclose*; entrambe le considerazioni però facevano fortemente perdere di accuratezza il risultato finale.

Nella versione finale dell'algoritmo invece, per tutte le 25 slice questo setup risulta sufficiente, ottenendo così un'ottima accuratezza.

Lo step successivo è quello di selezionare i pixel interni identificati dai bordi. Saranno i pixel corrispondenti al ventricolo. Questa semplice operazione è fatta con *bwselect*, con l'accortezza di utilizzare l'immagine *negata*.

Infine, l'ultimo step prima di andare ad estrarre i contorni, è quello di chiudere i possibili buchi contenuti nel ventricolo, smussando ed omogeneizzando lo stesso. Questo è realizzato tramite *imclose* e data la forma del ventricolo si è scelto un elemento strutturante "disco". Da questa mappa binaria che rappresenta i pixel del ventricolo sono poi state fatte le estrapolazioni richieste dall'esercizio.

Per l'elaborazione del LeftVentricle l'algoritmo è pressoché identico, anche se per ottenere una accuratezza migliore possibile è stata variata la parte di *imadjust* ed è presente una parte di *closing* dell'immagine (*imclose*) per omogeneizzare il risultato.

Un commento sul pixel di inizio image-reconstruction: dato che era richiesta minore interazione possibile da parte dell'utente, è stato scelto un pixel di inizio e lavorato sempre con quello. Avere sempre lo stesso punto di inizio ha consentito di massimizzare l'accuratezza dei bordi ricavati, con trasformazioni che probabilmente non hanno un riscontro generale sempre valido, per ogni scelta di inizio. In realtà a fronte di una riduzione dell'accuratezza è possibile, con questi stessi step, ottenere una validità abbastanza generale. Questa assunzione è motivata prima di tutto dal fatto che la struttura di codice per ventricolo destro e sinistro è la stessa, ma anche dal fatto che varie rifiniture e condizioni più "stringenti" dell'algoritmo, sono state impostate solo dopo aver ottenuto un risultato di bordi già, almeno in parte, soddisfacente.

