ITK

31-06-2012

È un linguaggio ad oggetti: INSIGHT TOOLKIT (è una libreria)
↳ supporto apposto per la segmentazione di immagini mediche
È solo un motore di calcolo, non permette la visualizzazione di immagini
→ VTK: supporto alla visualizzazione (VISUALIZATION TOOLKIT)
È diventato uno standard de facto (col tempo, senza che fosse imposto)

- È molto facile mettere il codice ITK in C++
- Wrapper → traduce le librerie di ITK in altri programmi (tipo matlab)
  → usando il linguaggio di matlab uso librerie ITK
- Processo in modo uguale 2D e 3D
- Molto efficiente: poche righe per molte operazioni
  → al massimo devo solo compilare i pezzi
- È uno strumento multipiattaforma: senza modifiche il codice deve funzionare su qualunque sistema del mondo
  CMAKE: rende disponibile lo stesso codice in architetture diverse
  ↳ CROSS-COMPILATORE: mi faccio creare ciò che mi serve

Quando INCLUDO le librerie ITK in C/C++ esse prendono il comando → se voglio evocare istruzioni associate ad altri namespace devo dirglielo
- std :: COUT → così gli dico di usare l'istruzione cout nel namespace standard (che non è quello di ITK, che è diventato il principale)

- OGGETTO: la potenza sta nell'accoppiamento tra variabili e metodi
  ↳ all'interno può avere tante variabili e delle funzioni (il codice è inserito in borgo parte nell'oggetto che cmq deve essere il + versatile possibile)
- CLASSE: modello generico, posso derivare quello che voglio
  "typedef" → a partire da una CLASSE definisco un oggetto

ITK → immagine con matrice N-dimensionale dove N è un numero qualsiasi
Posso definire i pixel anche come vettori (con + valori all'interno ad es. se faccio una rm posso attribuire allo stesso pixel i valori di Ti e Tz)

Puntatori → sono anch'essi oggetti: memoria che punta ad un'altra memoria
new() creo il puntatore → riferito ad una classe
delete() elimino il puntatore

⇒ memory leak: perdita di memoria → se dimentico di disallocare la memoria (altrimenti la mantengo occupata)

→ smart pointers: sono dinamici. alloco e disalloco la memoria automaticamente

Struttura PIPELINE: gestisce il flusso di dati

setInput: passo un oggetto → x es. al filtro
setOutput: prendo l'uscita

ciò che fa il filtro è privato: non vedo e non posso modificare.

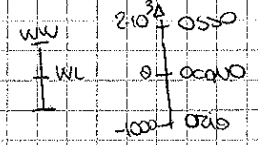
La PIPELINE è un collegamento simbolico tra puntatori

↳ l'ultimo blocco deve eseguire il comando di update per avviare la pipeline: si muove a ritroso finché non trova il punto di partenza (source: non ha nessun ingresso)

La pipeline è ora attiva → appena varia un parametro viene rievocato l'update e la pipeline viene rieseguita da dove ha avuto l'update in poi.

TAC → scala HU

$$\frac{\mu - \mu_{H_2O}}{\mu_{H_2O}}$$



corrispondenza HU: tono di grigio per in modo utile e usato 16 bit

WL → valore centrale
WW → ampiezza della banda

} perché non posso omni rappresentare tutta la scala Hounsfield