Iterative cone beam computed tomography in RTK, the Reconstruction ToolKit

Cyril Mory† and Simon Rit

iMagX project, ICTEAM Institute, Université catholique de Louvain, Louvain-la-Neuve, Belgium

Université de Lyon, CREATIS ; CNRS UMR5220 ; Inserm U1044 ; INSA-Lyon ; Université Lyon 1 ; Centre Léon Bérard, France

ICAR Group, LIRMM, Université de Montpellier, CNRS, France

Introduction

Lowering the imaging dose in 3D cone beam computed tomography (CBCT), either by reducing the number of projections or by reducing the dose per projection, is desirable in many clinical contexts. 3D + time CBCT is applicable to either cardiac or lung imaging, e.g. for interventional cardiology or lung radiation therapy. Both are challenging problems because of the small amount of projection data available, and the proposed solutions typically involve iterative reconstruction methods based on compressed sensing. Since these methods are computationally demanding, they have to be implemented efficiently and tested extensively on real data in order to become usable in clinical environments. This paper aims to introduce how such methods have been implemented within the C++ open-source tomography library RTK[1], which can handle Varian, Elekta, IBA, medPhoton and Precision X-Ray projection data out of the box.

Materials & Methods

The 3D tomography problem can be expressed, in terms of convex optimization, as the search for $\hat{f} = \underset{f}{\operatorname{argmin}} \|Rf - p\|_2^2$, where f is the sought 3D volume, R is the Radon transform or the X-ray transform, and p are the measured projections. A variety of iterative methods exists to find $\hat{f}$ in usual conditions: in RTK, the SART, OS-SART, SIRT, and conjugate gradient (CG) methods have been implemented for this purpose. When little data is available, the set of possible f contains undesirable solutions, and must be restricted by imposing additional conditions. Typically, f is requested to have a sparse gradient (total variation regularization) or a sparse wavelet transform. RTK contains implementations of the ADMM algorithm to perform these regularized reconstructions.

In 3D + time tomography, projections are acquired while the patient's body undergoes a periodic motion. A periodic signal describing that motion is either measured during the acquisition or inferred from the projection data (one such solution is available in RTK). Each projection is labelled with the phase of that signal, and each frame of the reconstructed 3D + time sequence represents the patient's body at a given phase. The 3D + time tomography problem can then be expressed as the search for $\hat{f} = \underset{f}{\operatorname{argmin}} \sum_\theta \|R_\theta S_\theta f - p_\theta\|_2^2$, where f is the sought 3D + time sequence of volumes, S_θ is an interpolator along the time dimension, R_θ is the Radon transform or the X-ray transform for a given projection, and p_θ is the measured projection at angle θ. For a given θ, S_θ interpolates between the two frames closest to the phase at which p_θ has been acquired. Various methods have been implemented in RTK to minimize this cost function: 4D SART, 4D OS-SART, 4D SIRT and 4D conjugate gradient. Adding regularization, both in each frame and between frames, yields the 4D ROOSTER method, also available in RTK. When some prior knowledge about the patient's breathing motion is available, for example when a displacement vector field (DVF) can be extracted from a prior 4D CT scan, that DVF can be used in 4D ROOSTER to improve the reconstruction.

All these iterative reconstruction methods are implemented by assembling simple processing units called "filters". In RTK, the forward and back projection filters, as well as many others, have been implemented in CUDA. Intermediate images are transferred automatically from host to GPU and from GPU to host, only when necessary, thanks to a class itk::CudaImage. Wherever possible, the filters are re-used from one application to the other: all the presented methods can use any of the available forward and back projectors, all methods based on CG (3D and 4D conjugate gradient, ADMM, ROOSTER) share the same CG filter, ROOSTER can use the same wavelets denoising filter as ADMM, all 4D methods share the same S_θ interpolator, etc…

Results

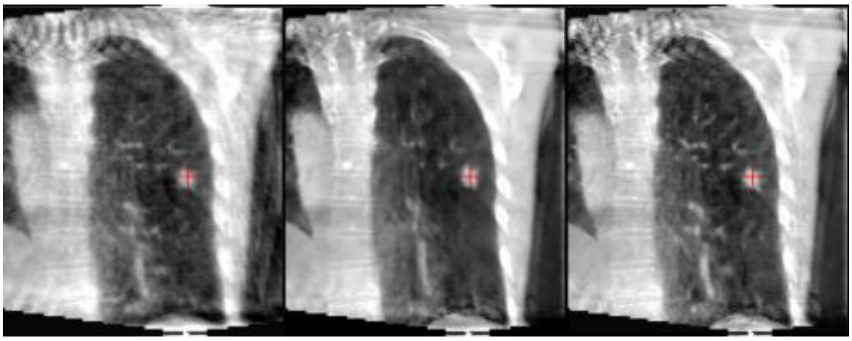

Figure 1: *Reconstruction results with several 4D iterative reconstruction methods available in RTK. On top: sagittal cuts, on the bottom: coronal cuts, from left to right: 4D conjugate gradient, 4D ROOSTER, motion-aware 4D ROOSTER*

Reconstructions of a sequence of 10 volumes of 220×280×370, from 656 projections of 512×512 pixels, on an Intel Xeon E5-2620 CPU with 12 cores, equipped with an nVidia GTX780 GPU, took 3 minutes for 4D conjugate gradient, 18 minutes for 4D ROOSTER, and 31 minutes for motion-aware 4D ROOSTER

Discussion & Conclusions

RTK efficiently implements many iterative reconstruction algorithms for both 3D and 4D CBCT and can readily be tested by medical physicists on a variety of CBCT scanners used in radiotherapy. Researchers can add new algorithms with little effort by assembling existing filters.

References

[1] S. Rit, M. Vila Oliva, S. Brousmiche, R. Labarbe, D. Sarrut, and G. C. Sharp, "The Reconstruction Toolkit (RTK), an open-source cone-beam CT reconstruction toolkit based on the Insight Toolkit (ITK)," in *Proceedings of the International Conference on the Use of Computers in Radiation Therapy (ICCR)*, 2013.

†Corresponding Author: cyril.mory@uclouvain.be