Rapport

Autor : Agata Krasoń

1. Manual rigid registration
   a) ***vv ct.mha cbct.mha***

      Slice presentation changes in axial, coronal, sagittal, horizontal and vertical plane.
   b) ***vv --linkall ct.mha cbct.mha***

      This command allows to align two images into one common coordinate system. But in this case it is not the best match.
   c) ***vv ct.mha --overlay cbct.mha***

      This is result of registration ct.mha and cbct.mha images in each presentation as coronal, axial, sagittal. This is not perfect match. We could see a little shift between projection. It is possible to improve registration by using image tool to manually registration.

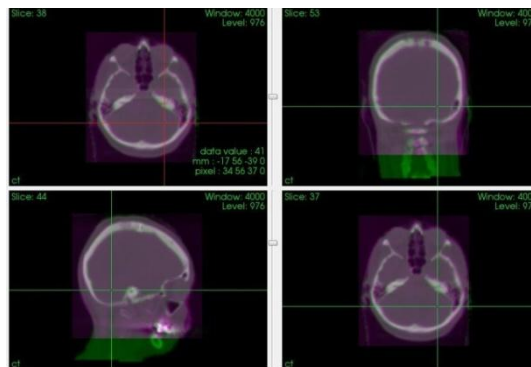   d) **Results manual registration in vv software**

Fig. 1. Results of manual rigid registration.

Gimble lock is the problem with image rotation between axis. During image rotation into one axis, other axis rotate also in the same time. It causes that one degree of freedom is lost. Image could not rotate correctly.

In result matrix has size 4x4 and it consists matrix of rotation and vector of translation. Example :

0.9996 0.0174 0.0177 -9.6840
-0.0177 0.999 0.0171 10.409
-0.0174 -0.017 0.999 13.082
0.0 0.0 0.0 1.0

Translation, Rotation x , Rotation y, Rotation z

Sometimes It is better to use quaternions as representation of angles instead of Euler angles. It allows to avoid gimble lock problem.

2. Automated rigid registration

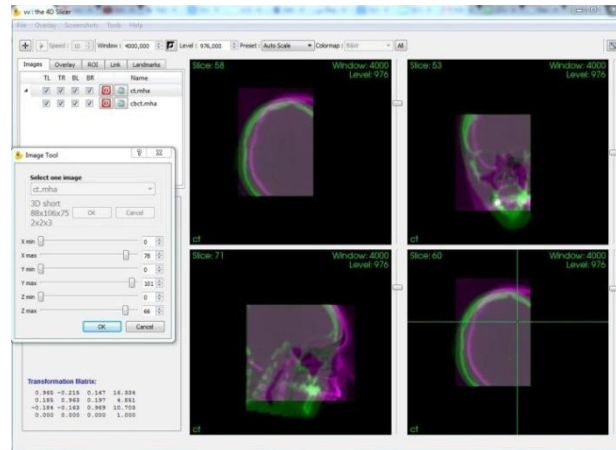*elastix -f cbct.mha -m ct.mha -p Par0005.MI.rigid.txt -out rigid*

Fig. 2. Visualization results of automated registration in vv.
Regions with not perfect match.

Different similarity measured :

Mutual information (*AdvancedMattesMutualInformation*) - final metric : - 0.969

Normalized mutual information (*NormalizedMutualInformation*) - final metric : -1.2975

Normalized correlation coefficient (*AdvancedNormalizedCorrelation*) – final metric : - 0.8118

3. Non –rigid registration

   a) *vv ct.mhd --overlay cbct.mha*
   b) *elastix -f cbct.mha -m ct.mhd -p Par0005.MI.1.txt -out dir*
      transformix -in ct.mhd -out dirres -tp dir/TransformParameters.0.R1.txt -def all
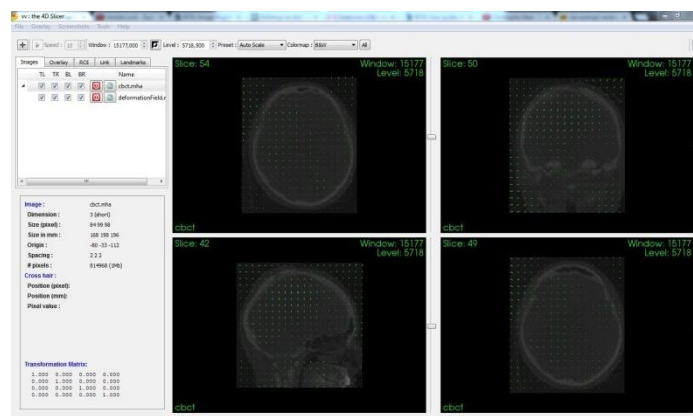   c) *vv cbct.mha --vf dirres/deformationField.mhd*

Fig. 3. Presentation of deformation field in vv software.

This command allows display register images with deformable field. Deformable field shows direction of local deformation vectors in the grid. It can be possible to see changes of local deformation in the image.

d) vv --linkall cbct.mha --overlay ct.mhd cbct.mha --overlay dirres/result.mhd
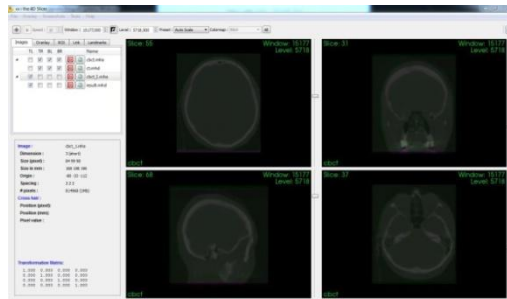

Fig. 4. Presentation of non rigid registration results in vv software.

This command allows display images after alignment and non – rigid registration with parameters from elastix. It can possible to see results of non – rigid registration based BSpline.

4. Modification of vv software. Automation loading parameters

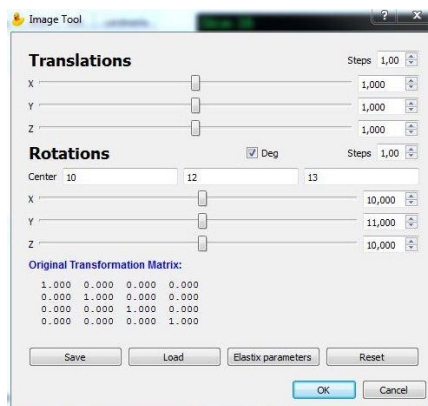Add button **"Elastix parameters"** in manual registration -> image tool :

Fig. 5. Modified image tool GUI.

Add new method ***LoadElastixFile()*** – allows to read rotation, translation and center of rotation parameters. These parameters could be apply to manual registration.

5. Conclusion

Rigid registration is global transformation. Rigid transformation contains only translation and rotation. It is very basic approach. It is very useful to register rigid object. In medical imaging it has to deal with local deformation. Especially in surgical and radiotherapy application soft tissue moves a lot (structures : lungs, liver, kidney) with heart bit or breathing motion. It is better to use non – rigid registration like : BSpline, Demons or TPS.

It will be good idea to join and integrate these two tools elastix and vv. **Elastix** is very good tool for registration but there is not any possibility to visualize results. On the other hand **vv** is useful tool to visualize registration results, 2D, 3D and 4D image sequence. These two software could use to analysis medical image data set. It will be easier for clinicians and all users. **VV** software functionality could be also extended by integrating with other registrations tools like : **IRTK** or **VTK CISG** or **Matlab toolbox FLAIR** .