**Matching of fiducial lines to slice intersection points in ultrasound images**

Alexis Boucharin, Andras Lasso, Csaba Pinter, Tamas Heffter, Gabor Fichtinger

*Laboratory for Percutaneous Surgery, School of Computing, Queen’s University, Kingston, Ontario, Canada*

***Purpose***

Ultrasound imaging devices are now part of prostate brachytherapy planning and delivery. Therefore, imaging systems have to be calibrated and tested as precisely as possible as it is important to avoid too high dose to preserve healthy tissue and too low dose which would not be sufficient to treat the tumor. Current image quality assurance methods are manual and therefore lengthy and operator dependent. To avoid these issues an automatic quality assurance method is necessary. However, currently there is no phantom that is suitable to perform all the required tests and research and development is needed to create new phantoms. Geometric calibration of ultrasound imaging systems requires phantoms and even though many versions are available, none of them is perfect, so more research and development work is required. Automatic fiducial line segmentation is therefore an important part of image quality assurance and calibration of ultrasound imaging systems. However, the current methods need to develop new code specific to a phantom. This operation is time consuming and the tolerance parameters are often fixed and obtained via a trial and error process. We propose a method that helps this research and development work by not requiring any software changes when using different phantom geometries. by handling several configurations for the fiducial lines, this method will detect coplanar lines that be contained in multiple planes for any number of lines per plane as well as any number of points per line. This method will also determine the tolerance parameters automatically from inputs such as maximum angular movement.

***Method***

The fiducial segmentation algorithm by (Chen, 2009) provides a list of fiducial points sorted by intensity that the method takes as an input.

From the list of fiducial points, N-point lines are computed and sorted by their intensity so that we have a list of lines each made of N fiducial points. The number of fiducial lines (fiducial points in a cross plane) is provided in the phantom definition file, as well as their basic structures such as parallel fiducial lines and Z-shaped fiducial structure. The number of these structures is not limited and is provided in the phantom definition file. Then, a backtracking algorithm is performed on the N-point lines found previously to match the actual lines made from the fiducial points from the phantom definition file. The choice for a backtracking algorithm is its simplicity and the fact that there are not too many candidate lines so the computation time of this part of the method is not preponderant. Once the lines are correctly matched, we can determine from image orientation and a transform matrix the correspondence between the fiducial points we found the actual one and therefore register them to the labels provided in the phantom definition file. Here is an example of a phantom definition file:

<PhantomDefinition version="1.0">

<Description

Institution="Queen's University PerkLab"

Version="1.0"

Type="Double-N"

Name="fCAL"

/>

<Model

ModelToPhantomOriginTransform="

1 0 0 -15.0

0 1 0 10.0

0 0 1 -5.0

0 0 0 1"

File="FCal\_1.0.stl"

/>

<Geometry>

<!-- N wire definitions -->

<NWire>

<Wire Name="E3\_e3" EndPointBack="20.0 40.0 5.0" EndPointFront="20.0 0.0 5.0" Id="1"/>

<Wire Name="F3\_j3" EndPointBack="45.0 40.0 5.0" EndPointFront="25.0 0.0 5.0" Id="2"/>

<Wire Name="K3\_k3" EndPointBack="50.0 40.0 5.0" EndPointFront="50.0 0.0 5.0" Id="3"/>

</NWire>

<NWire>

<Wire Name="E4\_e4" EndPointBack="20.0 40.0 0.0" EndPointFront="20.0 0.0 0.0" Id="4"/>

<Wire Name="J4\_f4" EndPointBack="25.0 40.0 0.0" EndPointFront="45.0 0.0 0.0" Id="5"/>

<Wire Name="K4\_k4" EndPointBack="50.0 40.0 0.0" EndPointFront="50.0 0.0 0.0" Id="6"/>

</NWire>

</Geometry>

</PhantomDefinition>

The different thresholds to accept points on a line or to register a potential line to an actual one is computed by the algorithm instead of implemented directly after a trial and error process. From the angular maximum movements of the ultrasound probe, we can determine how far from the actual position the candidate line can be. This angular maximum movement provides the range in which the image can actually be, as the image plane might not necessarily be perpendicular to the fiducial lines due to user movements, or could be slightly rotated around one axis or the other. These angular parameters can be obtained from the phantom definition file and the input data and would allow an optimal choice of threshold parameters that are automatically determined for any line configuration in the phantom.

The method has been developed in C++ with the **Insight Segmentation and Registration Toolkit** (ITK) for portability, speed and robustness. A 3D Slicer module has been developed for visualization of input data and results. This module was extensively used for software debugging, testing, and creation of ground truth data sets for automatic testing.

The algorithm is tested every night using CDash, which provides a consistent tool for testing and analyzing different information about the method such as speed of computation over time and after each update of the code to see what changes improved or slowed down the computation time.

***Results and Discussion***

***References***

1. Chen, T. T. (2009). Chen, T.K., Thurston, A.D., Ellis, R.E., and Abolmaesumi, P. *Ultrasound in Med. & Biol, 35(1) pp. 79–93*.