Marker Registration

Assignment for CISC-472

We have patient who undergoes serial CT imaging scans, at different times. The purpose is to monitor tumor growth in the brain. In each scan the patients position is different. Your job is to register each scan to the first so called canonical scan. At time of the first can, we place small tattoos on the scalp so that for each scan we can place fiducial markers in the exact same spot. In each scan we have 4 such markers on the patient, always in the same reproducible position.

We assume the head is hemispherical, 25cm in diameter. Each scan will cover the full head. We want to have good target registration in the entire head. In each scan, the image slices are in the transverse (x-y) plane. For simplicity, assume the center of the coordinate system is in the center of spherical head. The top of the head is in the +z direction.

You will develop a method (or methods) for registration of consecutive head scans and validation of the methods through simulation.

Marker Placement

Design and implement placement for the markers on the head. Explain your choice, pros and cons.

Target Simulation

Design and implement a method to generate simulated random targets inside the head.

Registration

Design two different ways to achieve registration between consecutive volume scans using marker points. Explain the pros and cons for each method. Derive the math and constituent frame transformations.

Implement both methods in MATLAB.

Test both methods in a handful of ground-truth transformation examples, such as creating uniaxis translation or uniaxis rotation between the two scans.

Choose one method that you will be using for the remainder of this assignment.

Pose Simulation

Design and implement simulation of poses between consecutive volume scans. This should be done by leaving one scan in the canonical pose and randomly transforming the other scan by a known transformation. Set the maximum translational displacement along each axis at 5cm. Set maximum rotation about each axis at 15degree.

Fiducial Localization Error (FLE)

Explain the concept of Fiducial Localization Error (FLE) in rigid marker registration (definition, math, purpose, what and how factors in it, how you measure it, limitations, etc.).

Design and implement a simulation strategy to generate random FLE from a given maximum FLE. (Explain possible strategies and justify your solution.) Input: MaxFLE. Output: FLE vector.

Fiducial Registration Error (FRE)

Explain the concept of Fiducial Registration Error (FRE) in rigid marker registration (definition, math, purpose, what and how factors in it, how you measure it, limitations, etc.).

Implement FRE computation and analysis (min, max, mean, std). Input: marker points, Output: minFRE, maxFRE, meanFRE, stdFRE.

Target Localization Error (TLE)

Explain the concept of Target Localization Error (TLE) in rigid marker registration (definition, math, purpose, what and how factors in it, how you measure it, limitations, etc.).

Design and implement a simulation strategy to generate random TLE from a given maximum TLE. (Explain possible strategies and justify your solution.) Input: maxTLE. Output: TLE vector.

Target Registration Error (TRE)

Explain the concept of Target Registration Error (TRE) in rigid marker registration (definition, purpose, what and how factors in it, how you measure it, limitations, etc.).

Push your system from maxTLE=0 to maxTLE =5mm, in 1mm steps.

Push the system from maxFLE=0 to maxFLE=5mm, in 1mm steps.

Design TRE analysis strategy composed from your modules. Explain the nesting of simulation cycles and numbers of repetitions you chose. Use figures, block diagrams, etc.

Explain your findings between trends of TRE, FRE, TLE, FLE. (Use plots, block diagrams, flow charts, tables or whatever you may find useful to convey your thoughts.)

GENERAL RULES

- Read the online syllabus carefully for general instructions on the submission of assignments.
- Always explain how you solve a problem. Use drawings, math formulas, text, block diagram, pseudo code - anything that you find them appropriate to convey your ideas. I must know that you understand what you are doing and I must be able to follow your reasoning. Depending on the quality and depth of your reasoning and discussion or results you may pick (or lose) lots of points.
- Write proper header and richly comment your code. There is no such thing as too much comment. Good style and neatness will earn you valuable points. The lack of these will cause reduction.
- Use decimal digits sensibly and consider what is precision is practical for the given problem. Generally, resolution much finer than 1 mm is not practically achievable, so use 0.1 mm as your. Use integer or decimal point format in your outputs. No exponential number format!
- Test each module fully and construct several test cases with known ground-truth answer. Construct the examples “on paper”, explain the result that you expect, then run the example through the code and show that your program is correct – it produces the ground truth you pre-computed.
- Write a testing m file(s) for each module or problem.
- Capture the output, to show that your program does what it is supposed to do. Make plots whenever it is requested or makes sense. Add explanation text as you see it useful.
- Use MATLAB routines for recurring tasks.
- Submit the m files and the captured output file, as well as any drawing, or supplemental information you feel relevant.
- **Have fun!**