REGISTRATION EXPERIMENTS - CONTINUED

6th January 2005

1 Experiments

1.1 28122004

Code checked out and re-compiled successfully in baine@isbe. CS machines are switched off for vacation period.

1.2 Normal_MI-29122004-1 (on Baine)

Description: Relatively long group-wise NRR optimisation using MI and the new code committed before Christmas.

Results: As bad as before, but no apparent changes were made to MI code.

1.3 Normal_Res_Info-29122004-2-5 (on Baine)

Description: Various parameters for res_info changed to see how histogram-based registration performs in 3-D. There were some issues with the way parameters were read from the text files.

Results: As bad as before. There is a need to look into code because arguments of res_info cannot be accepted.

1.4 Res_Info_Binning-30122004-1-2 (on Baine)

Description: Experiments that attempt the address the problem above and get information-theoretic registration to work in 3-D.

1.5 Res_Info_Binning-30122004-3 (on Baine)

Description: Trying group-wise NRR using res_info after rigid and affine alignment to see if it 'misbehaves'.

Results: The objective function completely 'misbehaves', still.

1.6 Res_Info_Binning-30122004-4 (on Baine)

Description: Attempting to see how varying the number of bins and filter width affects results, if at all. Bins size is 50, filter width is 3 compared with 256 and 5 respectively.

Intent: Seeing if there is a point in trying to change the above values and observe a working algorithm.

Result: Registration still badly degrades, with a different result.

Conclusion: red_info in 3-D needs to be reviewed at code level.

1.7 Res_Info_Binning-31122004-1 (on Baine)

Description: The resolution was increased (coarser) and the histogram values returned to their old state. Also, the number of knot-point was increased.

Result: The horrible results seen before are gone, maybe because of the change in resolution. More experimentation is needed.

1.8 Res_Info_Binning-31122004-2 (on Baine)

Description: To learn more about what happened in the last experiment, an NRR group-wise stage was divided into 4: level 2, 5 knot-points; level 2, 2 knot-points; level 1, 4 knot-points; level 1, 2 knot-points.

Intent: Seeing what it is that make the algorithm 'behave' and what it is that harms its operation.

Results: The exacerbation is no longer as bad, but it is there and it prevents, at all levels, registration from being reached. In fact, one of the two brains shrinks.

1.9 030105-1-2

Description: Simplex did not appear to terminate at the past so the problem is looked at again.

Comments: The algorithm terminates after a long time of NRR at coarse resolution with just 2x2x2 knot-points, 10 iterations. The output data is corrupted though.

1.10 030105-3

Description: To debug the above, one iteration of NRR was run, instead of 10.

Comments: failed again.

1.11 040105-1

Description: Trying to see if normal optimisation works and if current build is sane.

Comments: Unsuccessful run. The emphasis is now on fixing the error, not yet performing NRR optimisation using simplex or Powell which is the next step.

1.12 040105-2

Comment: The previous build was incomplete. This motivates the need to perform previous experiments again, e.g. simplex experiments.

Description: As previous one.

Results: The optimisation now works properly again (on the Computer Science domain)

1.13 040105-3

Description: With the problem gone, 030105-1 is attempted again.

Results: Quite an excellent step is made by simplex, even with only 2x2x2 knot-points!

1.14 040105-4

Description: Shorter version of the above (1 NRR iteration, with more knot-points though) to get a preview of expected results.

Results: The single iteration does tremendous work!

1.15 040105-5-6

Description: Long NRR using simplex and increasing number of knot-points. Suffix 6 is shorter (1 iteration at each NRR stages instead of 5)

Note: I suspect that number of iteration are not a valid argument for simplex in this implementation.

Results: Excellent results. Even larger optimisations are worth an inspection.

1.16 040105-7-8

Description: More knot-points in suffix 7, also another lower level in suffix 8. See above for more details.

Results: The result from suffix 7 show degradation due to the switch to comp_region_matcher from sequential_matcher. Suffix 8 aborted and changed to suffix 10 due to observations in suffix 7.

1.17 040105-9

Description: Like suffix 7, only with sequential_matcher.

Intent: Results are expected to supercede those from suffix 7. If that is the case, there's a need to find out to what extent (number of knot-points increase) results can be improved.

Comment: broken after many hours of work (system shut-down)

1.18 040105-10

Description: Like suffix 8, only with sequential_matcher.

Comment: Aborted due to scale of experiment (about 1 week on a Pentium 4).

1.19 050105-1

Description: Trying to work with Powell optimisation.

Results: rather good. Yet to be compared.

1.20 050105-2

Description: Work with downhill search again for comparisons.

1.21 050105-3

Description: Like 040105-5, but using Powell. Needed for comparisons.

1.22 050105-4

Description: Like 040105-5, but using downhill search. Needed for comparisons.

1.23 050105-5-8

Description: Simplex with varying tolerance. Early one is 0.01, latest is 0.00001 (increase by order of magnitude in subsequent experiment).

Intent: Seeing what tolerance is suitable for optimisation.

1.24 050105-9-12

Description: Powell with varying tolerance. Early one is 0.01, latest is 0.00001 (increase by order of magnitude in subsequent experiment).

Results: Identical in all cases.

Important: There appears to be a duration difference. Lower tolerance leads to a longer run.

1.25 050105-13-16

Description: Downhill search with varying tolerance. Early one is 0.01, latest is 0.00001 (increase by order of magnitude in subsequent experiment).

Results: Duration of optimisation seems just about identical and so are the results. Maybe the tolerance argument is meaningless in this context.

1.26 050105-17-20

Description: As above, but seeing the effect of changing n_per_dim. Values of 1, 10, 20 and 80 in order.

Results: Results are identical.

Conclusion: The argument of tolerance appears meaningless.

2 Next Stages

Implementation of 2 tools is necessary:

1) Given input volume and output filename (and some magnitude) apply NRR warps to the volume and save it.

2) Calculate trajectories of two warp lists and get mapping of points. Compare this with another trajectory or image and calculate a measure like SSD to get a measure of 'distance' from the correct warp.

Also to experiment with:

1) Get res_info (and later MI) working in 3-D.

2) Try to see how high a tolerance can be used to get equally good results. This can lead to a speed-up.