

Object_Tracking

- **Optimize / Move (Keyboard)** These check boxes let you specify which objects are allowed to move during the tracking optimization and moved with keyboard commands. DRRs for all of the objects selected in the *Object Configuration* widget will be generated, but only the objects selected here will be allowed to move. This allows you to perform hierarchical tracking, which can be useful when bones overlap in the X-ray images. For example, you can track the femur, then fix it in its optimized pose while tracking the patella.
- **Max Iterations** The maximum number of iterations for one pass of the optimization algorithm. For single-frame optimization, this is the number of iterations for one frame (reporting time). For 4D optimization, this is the number of iterations for the first pass. For the second pass of 4D, half of this number is used as the maximum.
- **Translation Ranges** There is a slider for each of the X, Y, and Z translation ranges. Each one controls the amount that the DOF is allowed to change from its initial value during optimization. X, Y, and Z are the axes of the X-ray lab reference frame (defined by the calibration object).
- **Rotation Ranges** There is a slider for each of the X, Y, and Z rotation ranges. Each one controls the amount that the DOF is allowed to change from its initial value during optimization. X, Y, and Z are the axes of the X-ray lab reference frame (defined by the calibration object).

Object Tracking Widget

- **Tracking** The range of reporting times which will be optimized when using the *Solve Range*, *Solve Poses in Range*, or *Solve Range 4D* commands.
- **Algorithm** The type of optimization algorithm to use. ASA (adaptive simulated annealing) is currently the only recommended algorithm. SPAN is another type of simulated annealing. LBFGSB is a bounded, least-squares algorithm (local optimizer).
- **Image Metric** The type of image metric [algorithm](#) to use when comparing the processed X-ray images to the processed DRR images. Changing the image metric will automatically update the image metric parameters widget. Currently there are two algorithms implemented: *Pixel Difference* and *Conditional Entropy*.
- **Solve Range** Starts the frame-by-frame optimization process. The selected objects' poses are optimized at each reporting time in the tracking range. It starts at the current time and proceeds towards the earliest, then goes back to the current time and proceeds towards the latest. Although each frame (reporting time) is optimized independently of the others, previously optimized frames will modify the pose map, which could affect the starting poses for subsequent frames. Right-clicking gives the option to add the frame-by-frame optimization to the job queue.
- **Solve Current Frame** Performs an optimization of the current frame. Right-clicking gives the option to add the current frame optimization to the job queue.
- **Solve Poses in Range** Performs an optimization of each reporting time in the tracking range that already has a pose specified in the pose map. Right-clicking gives the option to add the pose optimization to the job queue.
- **Solve Range 4D** Starts the 4D optimization process. The entire tracking range is optimized at

the same time. Right-clicking gives the option to add the 4D optimization to the job queue.

- **Stop** Stops the currently running optimization.
- **Evaluate** Evaluates the poses of the currently selected bones using the selected image metric. The results are written to the output window.
- **Poses Saved During Tracking** Sometimes the global minimum of the image fitness function is not the correct pose of the bone being tracked. This is often because the X-ray and DRR image processing parameters are not ideal (see *Image Optimization*, below). Other times, it is because the bone in the X-ray images is occluded by soft tissue or other bones, or because of inherent differences between X-ray images and DRRs (image resolution, X-ray scatter, CT thresholding, etc.). To account for these other times, the ASA tracking algorithm attempts to save bone poses that result in local minima of the cost function, while still searching for the global minimum. The idea is that if the correct bone pose is not the global minimum, it is likely to be a local minimum somewhere within the search range. The ASA algorithm thus attempts to save a set of local-minimum poses so that you can revisit them once optimization has finished. The *Max Poses* field specifies how many local minima to save. When the tracking optimization has finished, if the final bone pose is not the correct pose, you can re-apply the saved poses by dragging the slider for that bone. If one of them is correct (or at least better than the final pose), you can use the *Add Current Poses to Maps* command in the *Pose Map* widget to add that pose to the map, thus overwriting the final pose.

Note: When the ASA algorithm evaluates a particular bone pose during optimization, it cannot determine if it is actually a local minimum or not because it does not compute gradients. It estimates if the pose is a local minimum by examining its cost function value and its proximity to other poses with similar values. Here is an example that details the process. If you perform an ASA optimization of 1000 iterations, with *Max Poses* set to 20 and *Distance (squared)* set to 3.0, when the optimization is complete the 1000 bone poses that were evaluated are sorted from lowest to highest cost function value. The pose with the lowest value is marked as the global minimum and added to the list of saved poses. Then the pose with the next lowest value is checked to see how close it is to the first pose. The sum of the squares of the differences in the 6 DOF values is computed and compared to the *Distance (squared)* value (3.0 in this case). If the distance is less than 3.0, it is assumed that this pose is in the same valley of the solution space as the first pose, so it should not be saved as a local minimum. If it is greater than 3.0, the pose is added to the saved pose list. Then the next pose in the sorted list is checked. If it is not within 3.0 units of any pose already in the saved list, it is added to the saved list. This process proceeds until the saved list contains 20 poses. It will take some trial and error to determine the best value of the *Distance (squared)* parameter for a particular data set, but a value in the range of 1.0 to 3.0 is reasonable, as the solution spaces are often very bumpy with numerous local minima. The units for the DOF translations are model units (usually mm), and the units for the rotations are degrees.

From:
<https://has-motion.com/wiki/> - Wiki Documentation Site

Permanent link:
https://has-motion.com/wiki/doku.php?id=other:dsx:x4d:object_tracking&rev=1721157697

Last update: 2024/07/16 19:21

