Tracker plugin

The tracker plugin consists of the files plugins/tracker.plugin.R, plugins/tracker.conf.R, plugins/XXX.model.R, plugins/XXX.conf.R, where XXX is the name of a tracker model. So far, there are two track-building models:

- nearestneighbour: join closest neighbours into tracks, with some restrictions
- multiframecorr: maximize the gain of a pathcover in a moving window of scans

How tracks are built

- tracks are built from blips. So if a target is not detected as a blip (in a given scan), it won't be part of any tracks (in that scan)
- tracks are only built when the **tracker** plug-in is loaded and enabled
- currently, track building uses only these properties of a blip:
  - **x, y, z**: the coordinates of the blip centroid, in metres [Done: ~~TODO: this will~~ depends on antenna characteristics]
  - **t:** the timestamp of the blip centroid, in seconds. This is the timestamp of the start of scan, plus additional seconds corresponding to the azimuth of the centroid.
  - **area:** the area of the blip, in metres squared [Done: ~~TODO: this will~~ depends on antenna characteristics]
  - **int**: the area-weighted mean intensity of the blip, from 0.0 to 1.0; 1.0 means each sample in the blip has the maximum value possible for the radar hardware (e.g. 4095 for 12-bit digitization)
- a track is in one of three states:

  - **nascent**: the track has a single blip, and can be made active by adding a second blip
  - **active**: the track has two or more blips, and can be extended by adding more blips.
  - **complete**: the track is finished and no more blips can be added. A complete track *never* becomes active again.
    (except that, as described below in *Previewing a Scan*, a

track completed in a preview can be reactivated if you change track-building parameters and re-preview or process that scan)

- every blip is either added to a nascent or active track, or begins a new nascent track by itself
- a **tracker model** is an algorithm that assigns new blips to active tracks, and hence also decides
  which new blips should start their own new tracks (namely, those blips not assigned to tracks)
- if a nascent track has not had a second blip added to it **soon enough**, then it is discarded.
- **soon enough** is a user-specified parameter known as "*how long a lone blip is retained as a possible track starter*".*
- if an active track has not had a blip added to it in a **long time**, it becomes complete
- a **long time** is a user-specified parameter known as "*how long a track stays active after its last blip*".*
- if a complete track does not have **enough blips**, it is discarded
- **enough blips** is a user-specified parameter known as "*minimum number of blips required for a track*".*.
- if adding a blip to a track would imply the target moved **too fast**, it is forbidden
- **too fast** is a user-specified parameter known as "*maximum speed of tracked objects*" .*

** : these parameters are global: they apply across all tracker models. Particular models can apply their own criteria for when to start, extend, complete, and discard a track.*

How tracks are plotted

- to see tracks, you must check the "*Use slow tk plotting...*" ** box in the *radR display options* window
- tracks are displayed as polygonal arcs ("Tk canvas line items") in the plot window
- nascent tracks are shown using the **nascent track symbol**, which defaults to a diamond,
  but can be changed in the file plugins/tracker/tracker.conf.R; this allows you to more easily see a single point.
- active tracks are displayed in one colour, complete tracks in another (dimmer) colour
- an active track is only displayed after it is **big enough**

- **big enough** is a user-specified parameter known as "*minimum number of blips before a track is plotted*"**
- complete tracks are displayed if their last blip was not **too long ago**
- **too long ago** is a user-specified parameter known as "*how long to retain plots of complete tracks*"**
- if you move the mouse over a track, it is highlighted by colouring, thickening, and displaying an arrow head at the last point, indicating its direction (this requires that the pointerinfo plugin be enabled)
- if you left-click the mouse on a highlighted track, the data for the track are assigned to the global variable *tr*, which you can examine in the console window. Also, the track contents are copied to the clipboard.
- tracks are panned, zoomed, and rotated along with the rest of the plot. These things all happen faster if you turn off display of "*Blip*"*,* "*Hot*"*,* "*Cold*" and "*Other*" items, since then no scan-conversion is done.
- in Tk plotting mode, you may notice that when you zoom, pan, or rotate, the blips don't get zoomed/panned/rotated as quickly as the tracks. This is purely an aesthetic issue, and does not affect track building (or track display, once the blips have caught up).

**: *these parameters only affect track display, not track building.*

Previewing a scan

Just as it does for blip-finding, radR tries to show you how it is building tracks by providing a preview
of what it will do for a scan, and by letting you change relevant parameters to see how they affect this.
If you pause while the tracker plugin is enabled, the tracks and blips you seen on the screen are a preview;
i.e. they have not been "committed" by saving them to an output file or using them to update cell statistics.
Images of tracks have had new blips added to them, but this is just for display. If you change parameters for
blip-finding or track-building, you may see some track tails shift around, as different blips are assigned to different
tracks. For example, reducing the maximum object speed may separate some blips from tracks to which they

were assigned, because the implied speed during the last segment of the track is larger than the new maximum.
Tracks which are completed or discarded in a preview scan are shown in different colours, and this status is indicated
in the pop-up window when the mouse cursor enters them.

Each time you change parameters (whether blip-filtering parameters or track-building parameters) while paused or stopped, the scan is reprocessed (if necessary) and a new preview is shown. When you decide to proceed with the parameters you have set, the scan is once again processed with these parameters. So if you single-step (using the **Play-1** button) through an archive, each scan is processed (at least) twice, once for each preview and parameter change, and once "for real". If you play continuously through an archive (using the **Play** button), each scan is only processed once.

If you move through an archive manually by moving the player slider, each new scan you arrive at gets previewed.
The effect is to repeatedly process any existing tracks by trying to add blips from the previewed scan. These additions
are undone each time you preview a different scan. If there are no existing tracks, moving manually between scans
just outlines all blips with the nascent track symbol, since every blip starts a track.

Saving tracks to a file

There is now a **TRACK HOOK**, namely a function which is called for each track that is completed (and not discarded).
Currently, the *tracker* plugin uses this hook to write the track to a .CSV file.

Speed issues

The *tracker* plugin gui is implemented using R-level calls to Tcl/Tk. This approach, while allowing for rapid protoyping, carries significant runtime overhead. At some point, the tracker gui will be reimplemented in pure Tcl/Tk with a simple R API. radR really slows down when the number of blips per scan reaches 40 or so on an ongoing basis. You can improve speed slightly by unchecking the "*Update plot while playing*" and "*Use slow tk plotting*" boxes in the *radR display options* window, but then you won't get to see what's

happening. Turning off display of "*Blip*", "*Hot*", "*Cold*", and "*Other*" items but leaving "*Update plot while playing*" checked will hide blips but show tracks and is also somewhat faster,
since no scan conversion takes place.

The tracker in pictures

The following set of images shows the tracker operating on a sequence of scans taken by Carolyn Matkovich at Pubnico, Nova Scotia. The pictures were captured from radR by repeatedly pressing the **Play-1** button and doing a screen capture each time radR paused. When **Play-1** is pressed, the currently-displayed scan is processed, and the next scan is previewed.

./trackerPreviewOne.tgif.png

./trackerPreviewTwo.tgif.png

./trackerPreviewThree.tgif.png

./trackerPreviewFour.tgif.png

./trackerPreviewFive.tgif.png

./trackerPreviewSix.tgif.png

./trackerPreviewSeven.tgif.png

./trackerPreviewEight.tgif.png