



Features and Plugins

This page summarizes radR features, most of which are implemented as plugins. Not every plugin is loaded by default, but you can [load them manually](#) and/or have radR [load them automatically](#).

Data sources (*: requires 3rd party hardware)

- **antenna**: load, edit, and save antenna characteristics
- **seascan** (*): acquire data from a Rutter Inc. Sigma S6 digitizer card via Seascan server
- **seascanarch**: read raw radar data from .DAT files recorded by Rutter Seascan software
- **usrp** (New May 2012): acquire data from an Ettus Research USRP-1 with LFRX daughterboard and custom electrical front-end
- **xir3000** (*): acquire data from a Russell Technologies Inc. XIR3000 USB video processor board
- **xir3000arch**: read raw radar data from folders of .REC files created by RTI software
- **video**: process camera movies (e.g. radar display screenshots) in rectangular coordinates using radR's algorithms and interface
- **genblips**: generate artificial data from a simple, customizable dynamic model of target motion

Processing

- **batch processing**: [process multiple blipmovies](#) to extract tracks
- **custom blip filtering**: [write an R expression](#) involving blip parameters
- **remove persistent fluctuating clutter**: the [declutter plugin](#) works on data recorded as blipmovies
- **tracker**: assemble blips into tracks
- **zone**: define regions where blips are forbidden or filtered differently

Data Storage

- **blipmovie**: read and writes archives of blip data
- **rawarch**: read and write full scans of raw data

Visualization and Interface

- **bliptrails**: retain images of blips from some or all past scans in the plot window
- **console window**: evaluate R expressions while radR runs; has a simple [history mechanism](#).
- **gifmovie**: create animated .GIF files of radR in action
- **pointerinfo**: pop up a window describing what's under the cursor in the plot window
- **underlay**: allow a geo-referenced GIF image to be displayed under the plot window

Roadmap

These features are in the works:

- **more tracker models** - other models for building tracks; [perhaps the Kalman-filter-based Multiple Target Tracking and Identity Management algorithm of Hwang et. al.](#)
- **speed-ups** - some critical sections of code (including tracking infrastructure) need to be rewritten in C, C++, or tcl to get them up to real-time operating speeds under realistic conditions.
- support for **tilted t-bar antennas** (where the plane of rotation is not horizontal), including clipping the ground-echo portion of scans