



Zero-straddling pulses

radR operates on a scan (or "sweep") at a time. This means the data have a time discontinuity, or **cut**, at the north azimuth: data immediately to its left come from around 2.5 seconds later (for a radar rotating at 24 RPM) than data immediately to its right. Currently, radR ignores this and builds patches that straddle the cut, joining hot samples in pulse 1024 to those in pulse 1, say. This means that objects with bearings near north will appear either stretched or compressed, according to whether the longitudinal component of motion points west or east. This may in turn affect whether an object has an apparent area within the range for qualifying as a blip. (It also currently affects the **pulsetime** calculated for a blip by the **saveblips** plugin: a zero-straddling blip will (incorrectly) have a pulsetime that makes it appear to come from near the south azimuth, since large and small pulse numbers are being averaged to a medium pulse number. For now, we treat this as a **slow-target approximation**. Fixed since (at the latest) June 30 2008; saveblips uses the table `RSS$patches`.)

The correct way to create zero-straddling pulses is to treat the dataspace as a circular staircase, rising through time, rather than as a sequence of flat discs. We would then join hot samples that are adjacent on this staircase, rather than adjacent on the disc we see by looking down on a 360-degree slice of staircase.

One approach is to rework radR to process data one quadrant at a time, only extracting blips for a given quadrant once its two adjacent quadrants are available. This is probably the most elegant approach, but would require considerable reworking of code.

Another approach is to carry along the last quadrant of the previous scan for calculations.