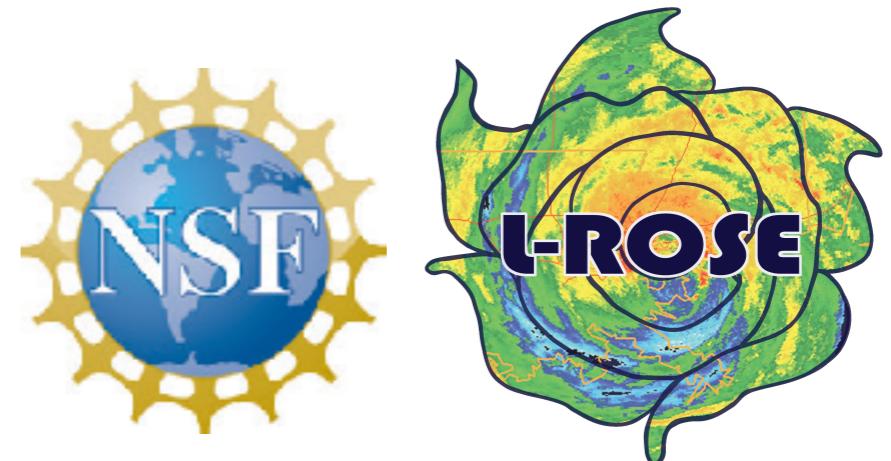


The Lidar-Radar Open Software Environment (LROSE) : Progress and Plans



Michael M. Bell
Colorado State University



Wen-Chau Lee and Mike Dixon
NCAR



LROSE: Lidar-Radar Open Software Environment

- LROSE is a project to develop common software for the LIDAR, RADAR and Profiler community
- Joint 4-year project between Colorado State University and NCAR Earth Observing Laboratory recently funded by NSF SI2-SSI (Software Infrastructure for Sustained Innovation)
- It is based on collaborative, open source development, with algorithms and analysis tools developed and supported by the community.
- Data stored in portable data formats such as CfRadial, based on UNIDATA NetCDF, following the Climate and Forecasting (CF) conventions to facilitate data assimilation by models



<https://nsf-lrose.github.io>



Infrastructure

NetCDF data support layer

Data servers for display applications

Higher-level language bindings for C++ library classes

Higher-level language native data handling library

Portability layer

Open source software distribution mechanism

Displays

ASCOPE for spectral radar I/Q time series

BSCAN for vertically pointing data

Low-level viewer and editor for polar data

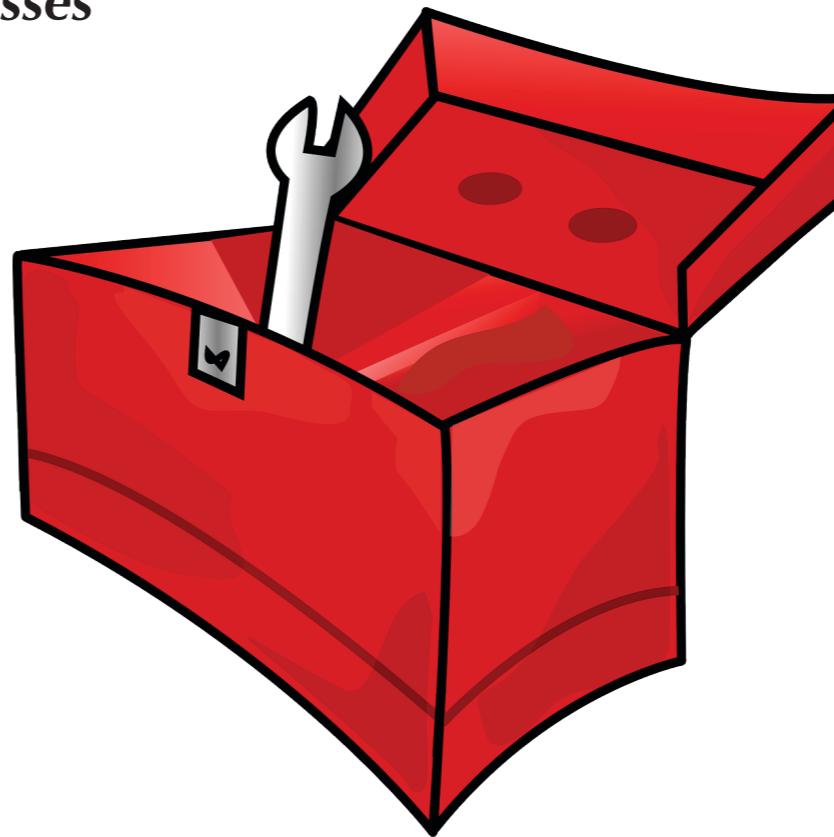
Platform-independent viewer for data integration

Display and editor tool for profiler data

Core Suite Algorithms

Well-tested, published algorithms

Common tasks and widely used tools



Community Algorithms

Specialized software toolsets

Configurable modules for different instruments and science goals

Partner-maintained packages (DOE PyArt, BALTRAD, and others)

April 11-12 Workshop Summary



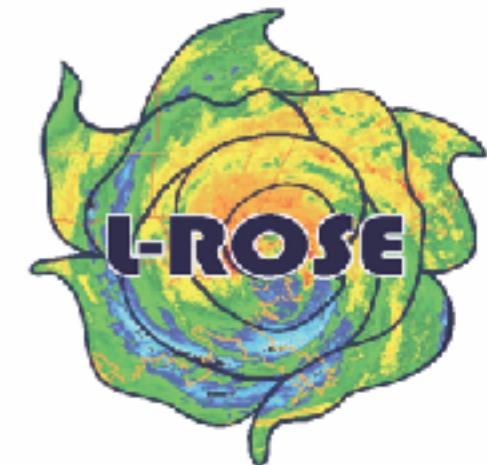
1. Six key applications were identified for initial development:
Convert, Display, QC, Grid, Echo, and Winds

2. Good documentation, ease of use, and starter kits were emphasized as crucial aspects for software adoption, not just the availability of code or algorithms

3. Community contributions beyond code such as discussion forums, how-to videos, and tutorials will help build user base

4. Focus on high-quality, well-tested, well-maintained and well-documented key applications as ‘building blocks’, allowing users to assemble trusted workflows to accomplish more complex scientific tasks

LROSE Convert



- **RadxConvert** (*mature prototype*)

- Convert to/from CfRadial netCDF format

Type	Notes	Type	Notes
D3R	Read-only	NEXRAD Level 2	Read-write
DOE	Read-only	NEXRAD Level 1,3	Read-only
DORADE	Read-write	NOXP	Read-only
EEC-Edge	Read-only (work in progress)	NSSL-MRD	Read-only
FORAY	Read-write	ODIM-H5	Read-write (work in progress)
Gamic	Read-only	RAPIC	Read-only
Gematronik Rainbow	Read-only (writeable with python script)	SIGMET Raw (Vaisala)	Read-only
HSRL (LIDAR)	Read-only	TDWR	Read-only
HRD (Hurricane Research Division)	Read-only	TWOLF	Read-only
Leosphere (LIDAR)	Read-only	UF	Read-write
CfRadial-1	Read-write	CfRadial-2 (WMO)	Read-write (work in progress)

LROSE Display



1. **Soloii** (*legacy*) has been ported to Docker

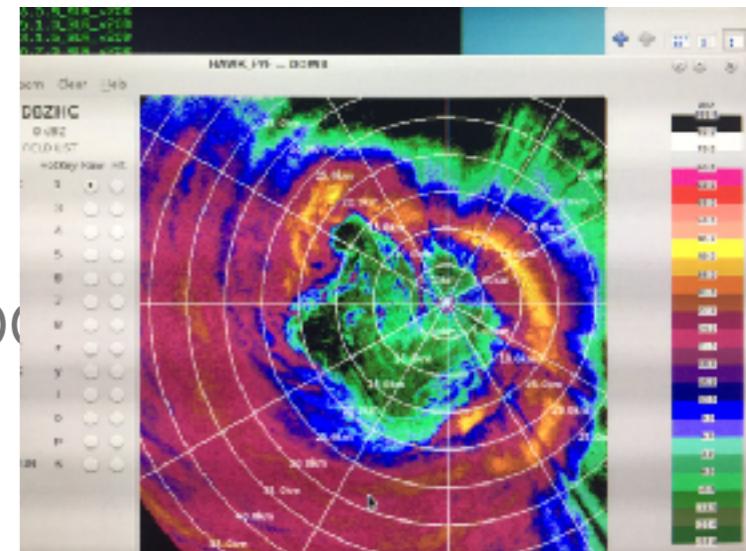


★ Download Docker and soloii_docker script. **Go!**

<https://github.com/NCAR/lrose-soloii/blob/master/docker/soloii-docker>

2. **HawkEye** (*in development*)

- Engineering and real-time display suitable for both scanning and vertically pointing radars. New research capabilities will be added



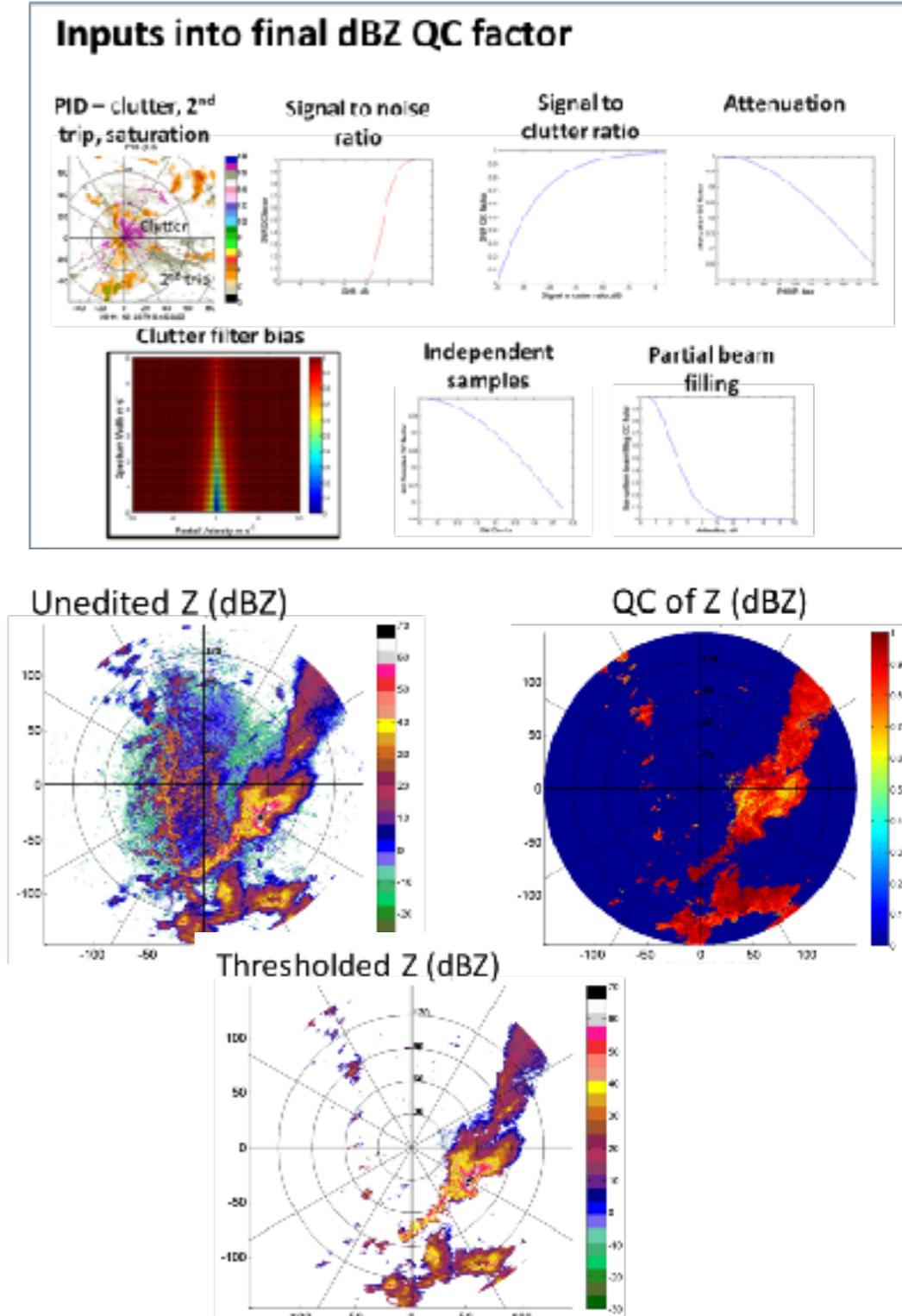
3. **Jazz** (*in development*)

- Java-based integrating display under active development (next generation to Zebra and CIDD)

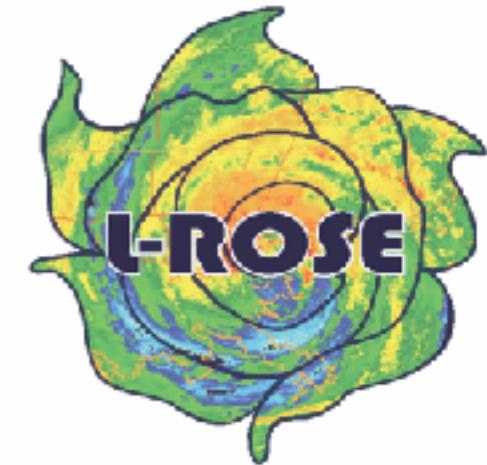
LROSE QC



1. **RadxQC** (*in development*): Remove AP & sea clutter, RF interference, chaff
2. **RadxModelQC** (*in development*): Compute QC factor for automated editing
3. **AirborneRadarQC** (*in development*): Airborne navigation corrections and automated editing
4. **JamesDealias** (*mature prototype*): 4D velocity dealiasing



LROSE Grid

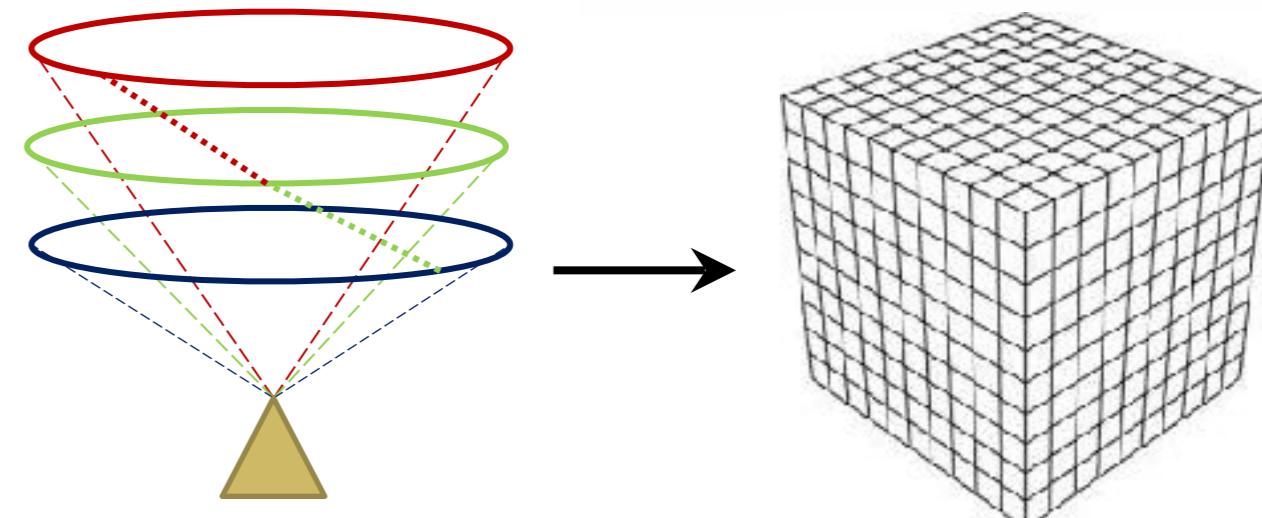


1. Radx2Grid (*mature prototype*)

1. 3-D Cartesian grid (x, y, z)

2. Cartesian PPIs (x, y, elev)

3. Regular polar grid (range, az, elev)



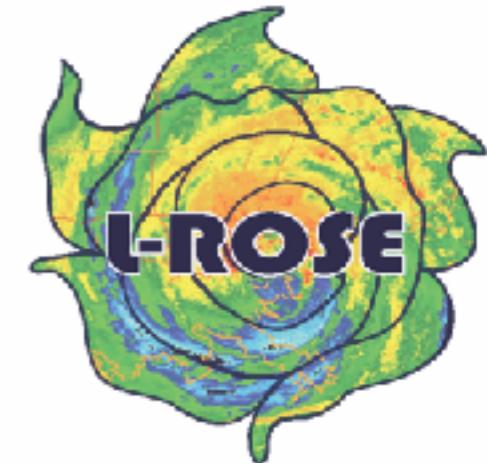
- Some users find Radx2Grid difficult to use due to many options in the parameter file. Development aimed at improving ease of use, documentation, and starter kits

2. Sprint & Reorder (*legacy to be ported to Docker*)



1. **RadxPartRain** (*mature prototype*): NCAR PID, Precip rate
2. **ConvStrat** (*mature prototype*): Convective-stratiform partition
3. **RadxBeamBlock** (*mature prototype*): Beam blockage identification
4. **QPEVerify** (*mature prototype*): Precipitation verification
5. **RadxQpe** (*in development*): QPE application
6. **Refract** (*mature prototype*): Refractivity calculations

Plan to consolidate and document with example applications



1. *Multi-Doppler*

1. **CEDRIC** (*legacy to be ported to Docker*): Well-tested and reliable for classic solutions, but not upgradeable.
2. **SAMURAI** (*mature prototype*): Spline-based 3DVAR solver
 - Inter-comparison with PyArt, HRD, and others beginning

2. *Single Doppler*

1. **RadxEvad** (*mature prototype*): Enhanced VAD
2. **VORTRAC** (*in development*): GBVTD

Progress and Plans



1. Year One of LROSE focused on building common infrastructure and evaluating community needs, Years 2 - 4 will expand our community software tools through open source development
2. Six key applications were identified for initial development: Convert, Display, QC, Grid, Echo, and Wind
3. Focus on high-quality, well-tested, well-maintained and well-documented key applications as 'building blocks', allowing users to assemble trusted workflows to accomplish more complex scientific tasks
4. Several mature prototypes available now, formal release of more tools in coming months via LROSE mailing list

<https://nsf-lrose.github.io>

Source is available!



- These are the current repositories for LROSE, which may change over time. Sign up for the mailing list to be updated

<https://nsf-lrose.github.io>

Description	GitHub URL
Main LROSE GitHub site	https://github.com/nsf-lrose
LROSE core	https://github.com/NCAR/lrose-core
LROSE NetCDF support libs	https://github.com/NCAR/lrose-netcdf
Color scales and maps for LROSE displays	https://github.com/NCAR/lrose-displays
Matlab display for CfRadial data	https://github.com/NCAR/lrose-emerald
Java-based Jazz display	https://github.com/NCAR/lrose-jazz
Code for legacy soloii application	https://github.com/NCAR/lrose-soloii
Code for legacy solo3 application	https://github.com/NCAR/lrose-solo3
Wind and airborne radar analysis	https://github.com/mmbell