

areaDetector: What's New?

Mark Rivers

GeoSoilEnviroCARS, Advanced Photon Source

University of Chicago



Moved areaDetector to github

- areaDetector was getting too big.
 - New releases being held up waiting for testing on one detector types, etc.
- Hard to collaborate with other sites using APS Subversion repository
 - git and github provide much better tools for multi-site collaborations
- Moved in December 2013, with R2-0 release in April 2014
- Split into 3 “core” repositories, and separate repositories for each detector

New Organization

areaDetector

Top-level module

RELEASE files, documentation, Makefile

ADCore

Core module

Base classes, plugins,
simDetector,
documentation

ADB binaries

Binary libraries for
Windows (HDF5,
GraphicsMagick)

ADProsilica

Prosilica driver

ADPilatus

Pilatus driver

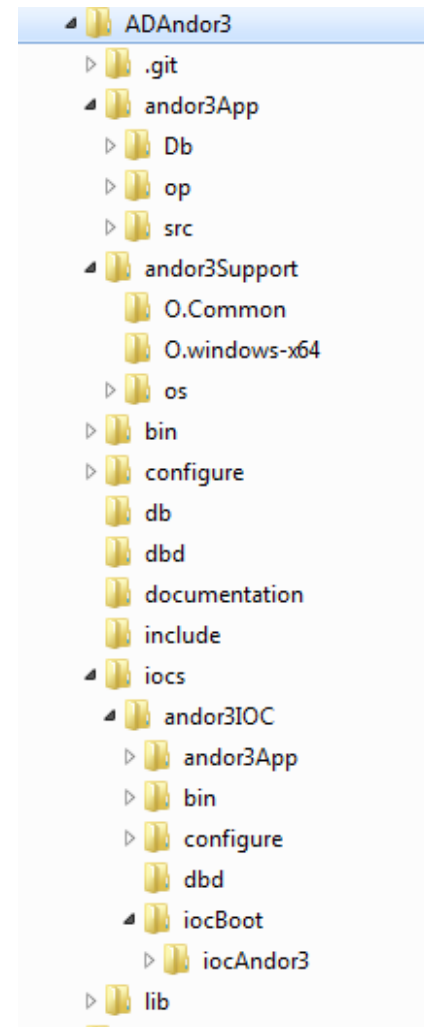
...

- Each box above is a separate git repository
- Can be released independently
- Hosted at <http://github.com/areaDetector> project
- Each repository is a submodule under areaDetector/areaDetector
- Can clone individual repositories, or clone the entire project with
`git clone -recursive https://github.com/areaDetector.git`

New Organization

Top-level areaDetector

Name	Date modified	Type
.git	10/17/2014 4:54 PM	File folder
ADADSC	10/17/2014 5:40 PM	File folder
ADAndor	10/17/2014 5:40 PM	File folder
ADAndor3	10/17/2014 5:40 PM	File folder
ADBinaries	10/17/2014 5:38 PM	File folder
ADBruker	10/17/2014 5:40 PM	File folder
ADCore	10/17/2014 5:39 PM	File folder
ADFireWireWin	10/17/2014 5:40 PM	File folder
ADLightField	10/17/2014 5:40 PM	File folder
ADmar345	10/17/2014 5:40 PM	File folder
ADmarCCD	10/17/2014 5:40 PM	File folder
ADPerkinElmer	10/17/2014 5:40 PM	File folder
ADPilatus	10/17/2014 5:40 PM	File folder
ADPixirad	10/17/2014 5:40 PM	File folder
ADPointGrey	10/17/2014 4:54 PM	File folder
ADProsilica	10/17/2014 5:40 PM	File folder
ADPSL	10/17/2014 5:40 PM	File folder
ADPvCam	10/17/2014 5:40 PM	File folder
ADQImaging	10/17/2014 5:01 PM	File folder
ADRoper	10/17/2014 5:00 PM	File folder
ADURL	10/17/2014 5:40 PM	File folder
aravisGigE	10/17/2014 5:02 PM	File folder
configure	10/17/2014 4:51 PM	File folder
documentation	10/17/2014 5:00 PM	File folder
ffmpegServer	10/17/2014 4:59 PM	File folder
ffmpegViewer	10/17/2014 4:55 PM	File folder
firewireDCAM	10/17/2014 4:53 PM	File folder
.gitignore	3/6/2014 2:13 PM	GITIGNORE File
.gitmodules	10/14/2014 6:45 PM	GITMODULES File
INSTALL_GUIDE.md	9/17/2014 5:02 PM	MD File
makeADPrebuilt	4/4/2014 3:18 PM	File
makeADPrebuilt_Current	4/4/2014 3:59 PM	File
Makefile	3/5/2014 1:57 PM	File
makePrebuiltAndor	4/15/2014 4:02 PM	File
README.md	3/26/2014 9:21 PM	MD File
RELEASE.md	10/15/2014 12:18 ...	MD File



- andor3App/ builds only a driver library, and depends only on base and asyn
- iocs/ builds an application and depends on autosave, busy, etc. Can disable building this.

New drivers

- ADAndor3 driver for sCMOS cameras from Andor
- ADPointGrey driver for GigE, USB-3.0, USB 2.0 and Firewire cameras from Point Grey
- ADLightField driver for Princeton Instruments cameras using their LightField application
- ADPixirad driver for CdTe pixel array detectors from Pixirad
- ADAndor added Shamrock spectrometer control for Andor cameras
- ADmarCCD added support for triggered acquisition using new high-speed (-HS) detectors from Rayonix
- ADPSL major rewrite of Photonic Sciences Limited driver to support new server features

R2-0: Point Grey driver

- New driver for all cameras from Point Grey using their FlyCap2 SDK.
- Firewire, GigE and USB 3.0
- High performance, low cost
- Example: Model GS3-U3-23S6M
 - 1920 x 1200 global shutter CMOS
 - No smear • Distortion-free
 - Dynamic range of 73 dB
 - Peak QE of 76%
 - Read noise of 7e-
 - Max frame rate of 162 fps (~400 MB/S, 4X faster than GigE)
 - USB 3.0 interface
 - \$1,295
 - Comparable to PCO Edge and Andor Zyla for 10X less money

Mono
Sensor

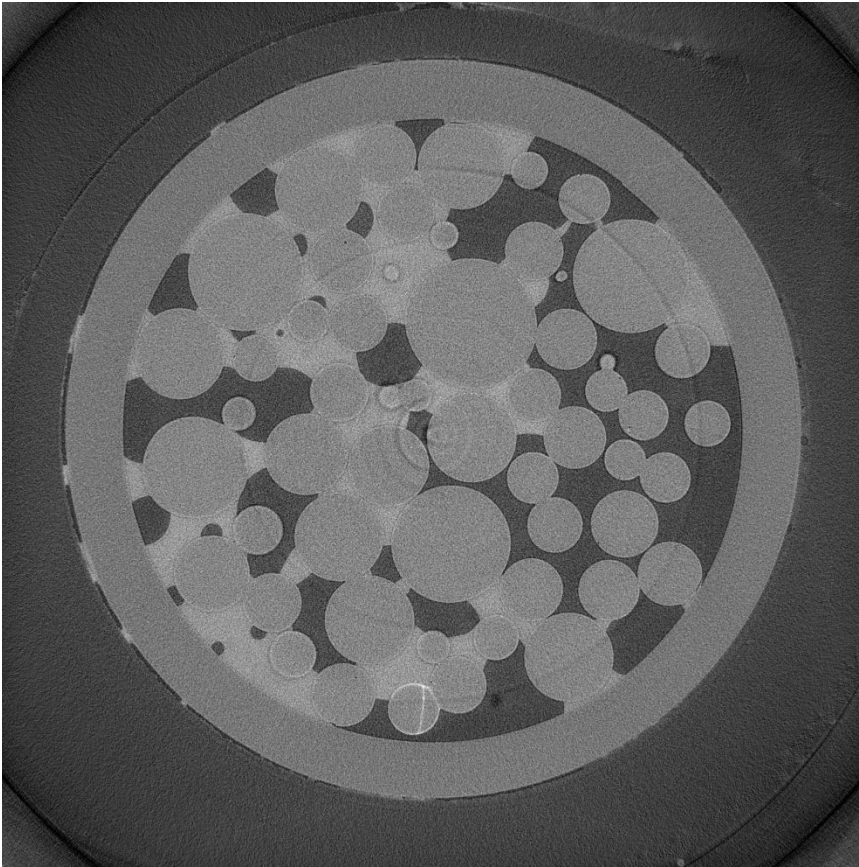
2.3 MP

USB
VISION

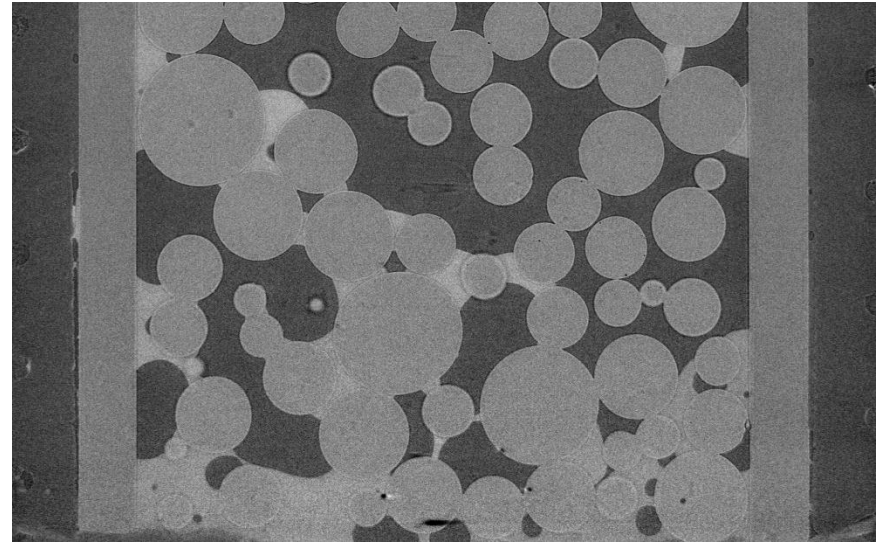


Pink Beam, Mirror=2.0 mrad

- Mirror angle=2.0 mrad (Beads_Pink_H)
- 2 mm Al absorber
- 8-bit data
- 1 ms exposure time, 124 frames/s, 900 projections, 7.3 seconds total
- Rotation axis orientation corrected for mirror angle



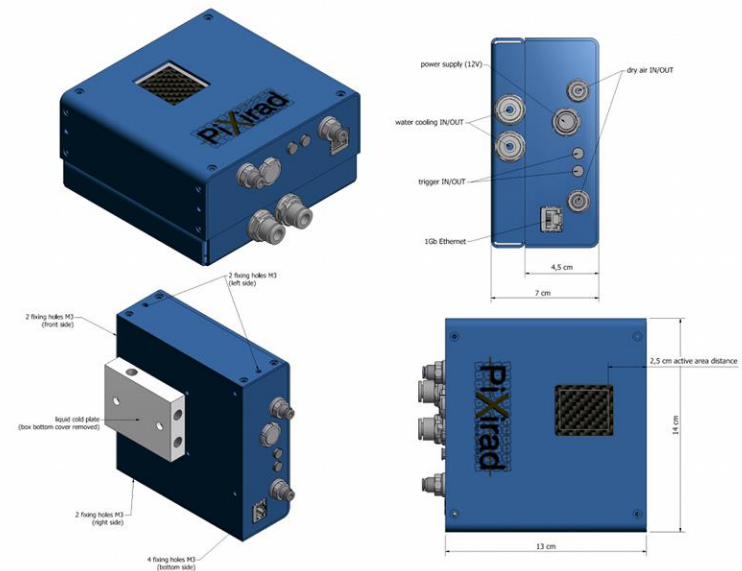
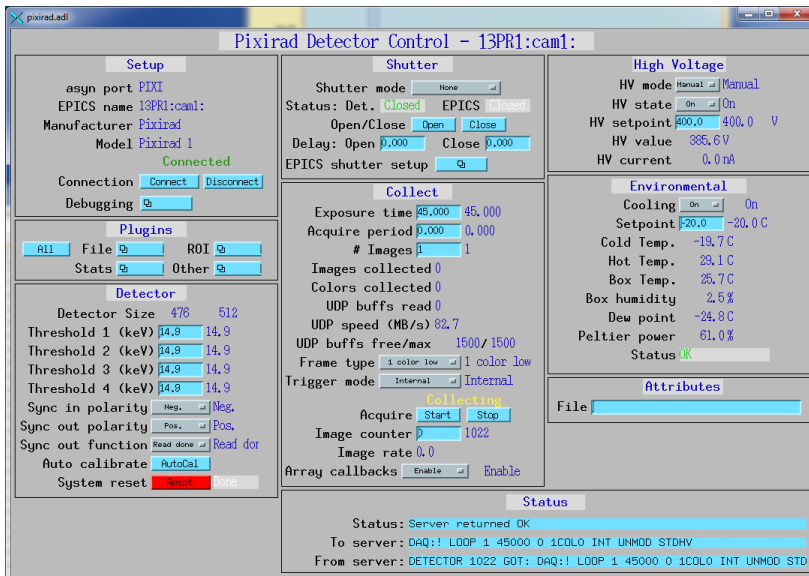
Horizontal slice



Vertical slice

R2-0: Pixirad driver

- New driver for Pixirad CdTe pixel array detector
- Similar to Pilatus, but CdTe gives very high efficiency to 80 keV or more
- 2 energy thresholds, so 2 different energy images simultaneously



Recent Additions (R2-0)

Improved timestamp support

- New timestamp functions added to `asynManager` in response to need from LCLS. These are used to set a user-defined timestamp source, to attach that timestamp to `pasynUser`. Standard asyn device support sets `prec->time` with this value, which will be used if `prec->TSE=-2`.

```
asynStatus (*registerTimeStampSource)(asynUser *pasynUser, void
    *userPvt, timeStampCallback callback);
asynStatus (*unregisterTimeStampSource)(asynUser *pasynUser);
asynStatus (*updateTimeStamp)(asynUser *pasynUser);
asynStatus (*getTimeStamp)(asynUser *pasynUser, epicsTimeStamp
    *pTimeStamp);
asynStatus (*setTimeStamp)(asynUser *pasynUser, const
    epicsTimeStamp *pTimeStamp);
```

- Added new virtual methods to `asynPortDriver` for timestamp support. These are `updateTimeStamp()`, `setTimeStamp()`, `getTimeStamp()`.

Recent Additions (R2-0)

Improved timestamp support

- Added new `epicsTimeStamp` field to `NDArray`. The previous `timeStamp` field was a double. Useful for plotting and math, but only microsecond precision and not useful for record timestamps.
- All drivers now call the following as soon as an image is collected:

```
updateTimeStamp (&pImage->epicsTS) ;
```
- `NDPluginDriver::processCallbacks` now calls the following when an `NDArray` is received:

```
setTimeStamp (&pArray->epicsTS) ;
```
- This means that all EPICS records, including plugin records, associated with a specific `NDArray` will have the timestamp when that array was collected if the record has `TSE=-2`

R2-0: Attributes

- Added new attribute type, `NDAAttrSourceFunct`. This type of attribute gets its value from a user-defined C++ function. It can thus be use to get any type of metadata. Previously only EPICS PVs and driver/plugin parameters were available as metadata.

File Plugin Enhancements

- NDPluginFile
 - Added new optional feature "LazyOpen" which, when enabled and in "Stream" mode, will defer file creation until the first frame arrives in the plugin.
 - Removes the need to initialize the plugin with a dummy frame before starting capture.
- NDFileTIFF
 - Supports any NDArray data type
 - Stores NDAttributes as ASCII user tags, up to 490.
- NDFileHDF5
 - Now supports using an XML file to define the layout and placement of NDArrays and NDAttributes in the HDF5 file.
 - Can be used to create NeXus-compliant files without using NDFileNeXus plugin
 - NDFileNeXus will probably be deprecated in a future release.

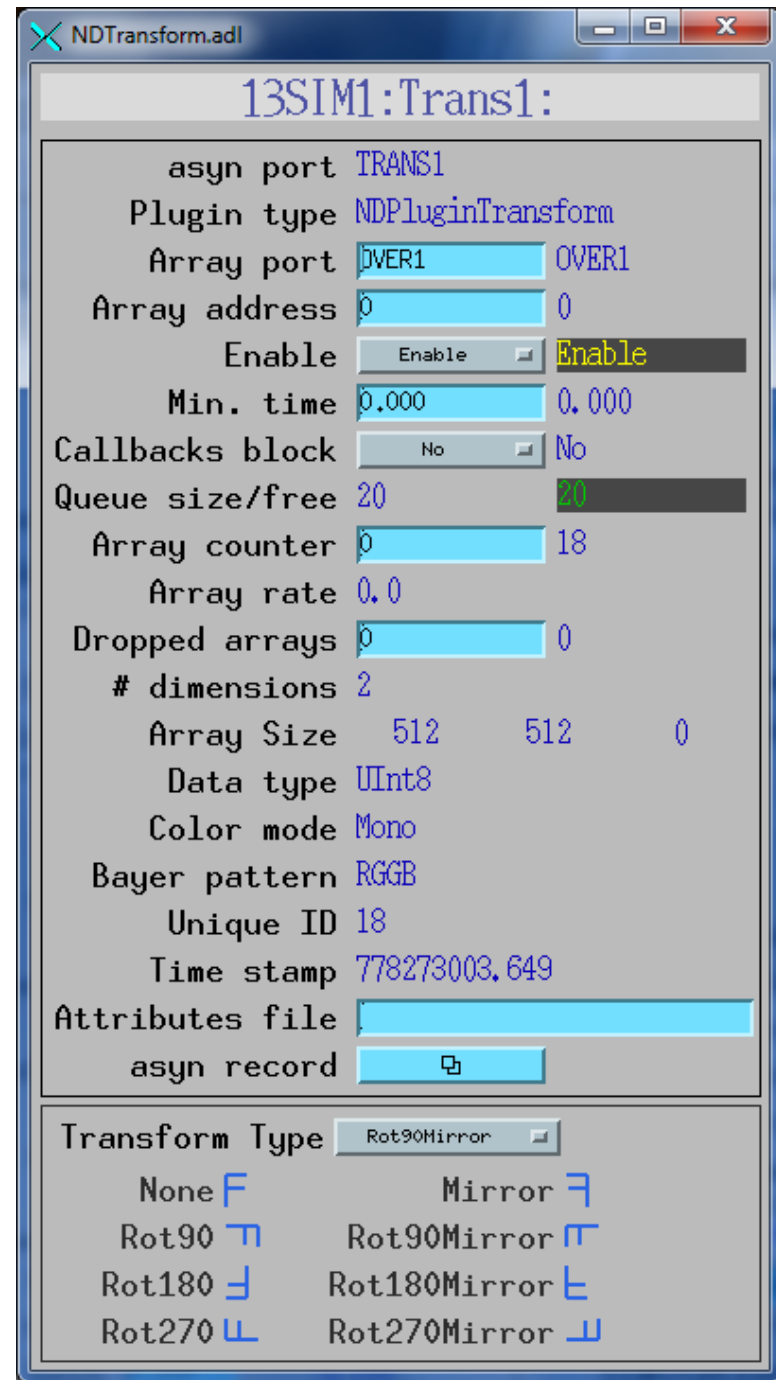
R2-0: Attributes

- Added new attribute type, `NDAAttrSourceFunct`. This type of attribute gets its value from a user-defined C++ function. It can thus be use to get any type of metadata. Previously only EPICS PVs and driver/plugin parameters were available as metadata.

Transform plugin

R2-1 changes

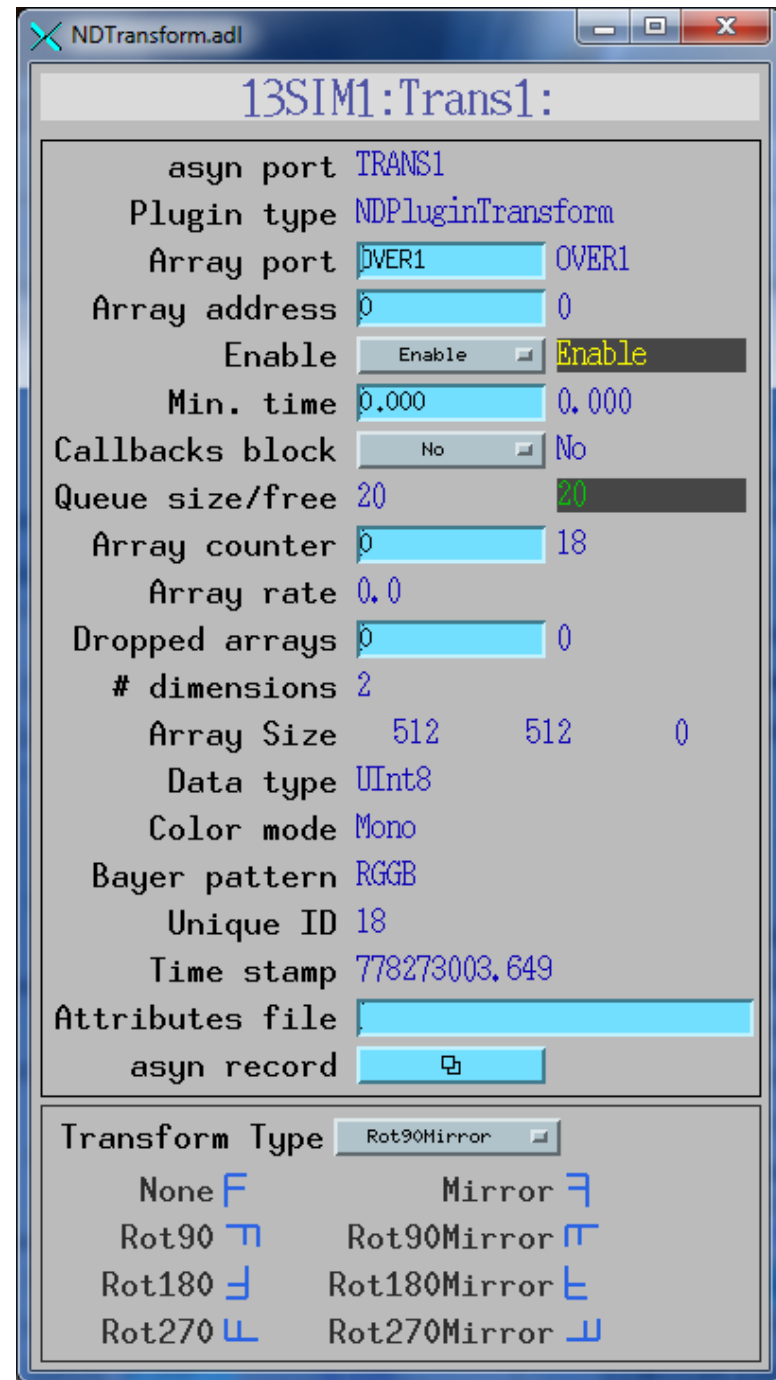
- Greatly simplified: just 8 operations including null operation
- 13-85 times faster than previous releases depending on data type, color mode



Transform plugin

R2-1 changes

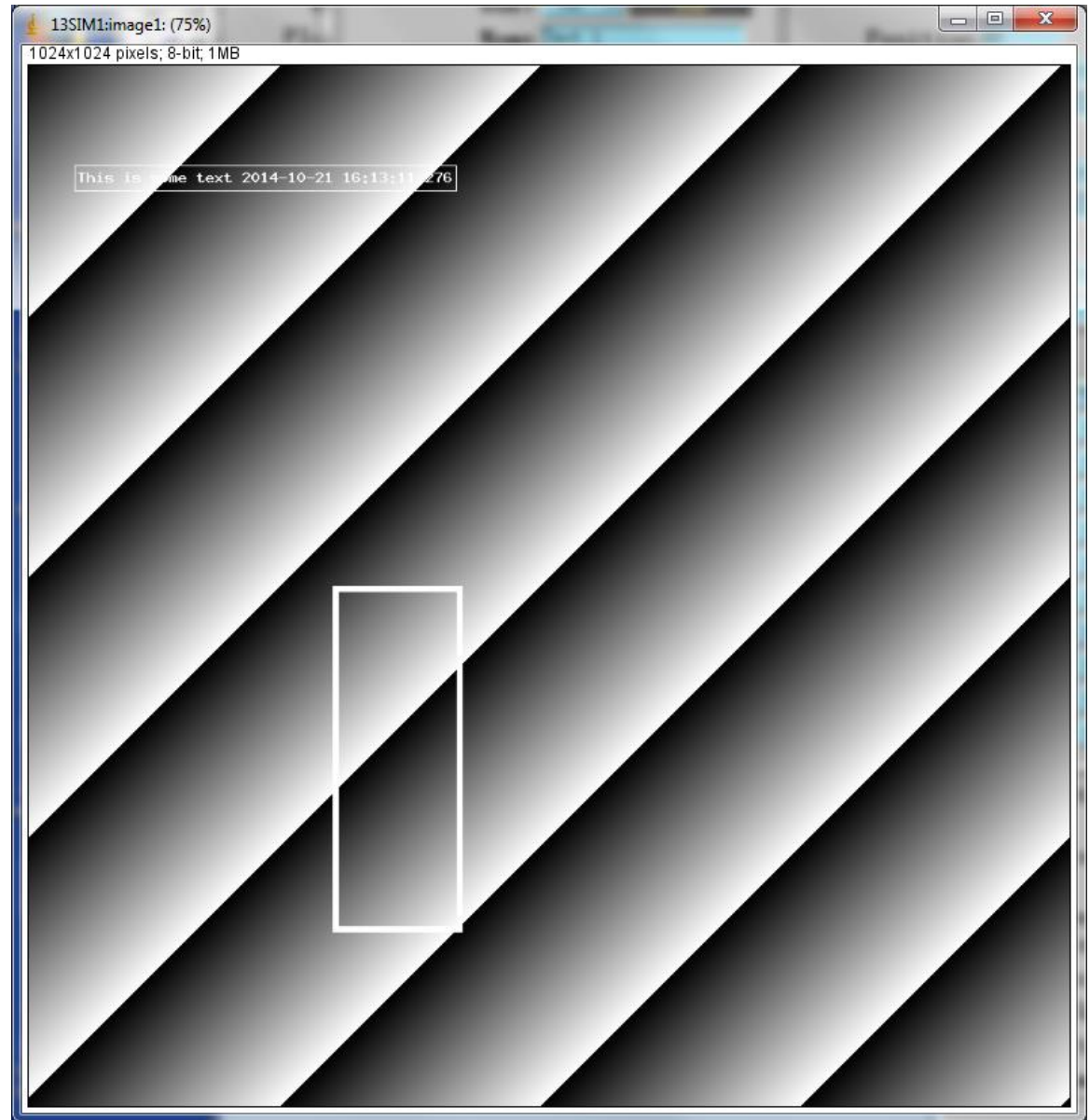
- Greatly simplified: just 8 operations including null operation
- 13-85 times faster than previous releases depending on data type, color mode



Overlay plugin

R2-1 changes

- Added support for text overlays
- Added support for line widths in cross and rectangle overlays



Future Ideas

- Put more functionality into ADDriver base class
 - Currently it does not do much, all code is in each driver for:
 - Doing callbacks to plugins
 - Processing new exposure time with writeFloat64 function
 - writeFloat64 in ADDriver base class would call setExposure() in derived class
 - Derived class would call ADDriver::doPluginCallbacks(), which would handle setting attributes, getting timestamp, calling plugins, etc.
 - This is the way the Model 3 motor driver, which also uses asynPortDriver, is written
- Demultiplexor & multiplexor plugins
 - Allow multiple plugins of same type to work on the same data stream when it saturates a single core

Future Ideas

- Extend areaDetector concepts to other types of detectors:
 - ADCs
 - Electrometers
 - Waveform digitizers
 - Oscilloscopes?
- They all produce 1-D (or 2-D for multi-channel inputs) arrays that could benefit from plugins for file saving, FFTs, ROI extraction, digital filtering, etc.
- Very impressive V4 features David Hickin presented yesterday
- These ideas will be discussed at workshop at Diamond next week