Brookhaven National Laboratory

U.S. Department of ENERGY

EPICS controls for xrt and integration with bluesky

Roman Chernikov (DSSI-DAD)

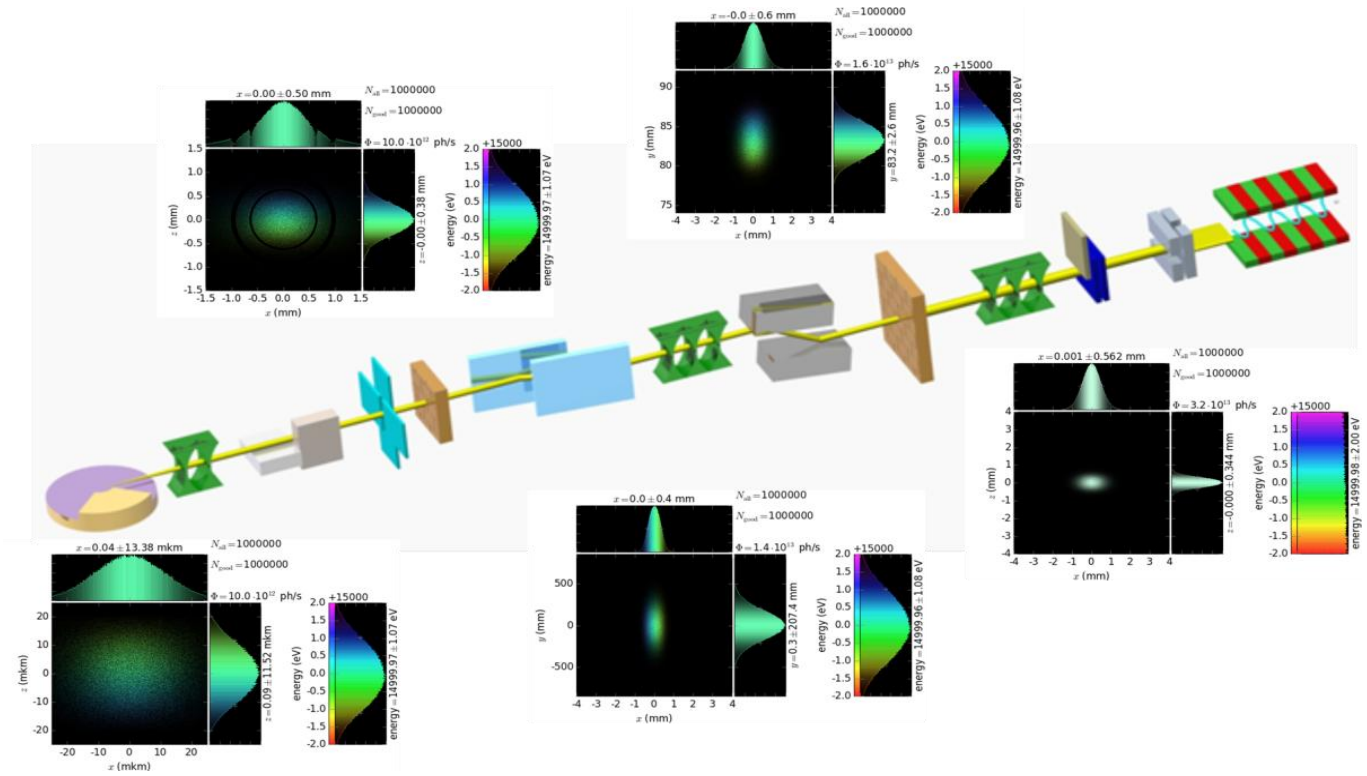
May 16 2025

@BrookhavenLab

xrt – open source, python-based optical simulation tool

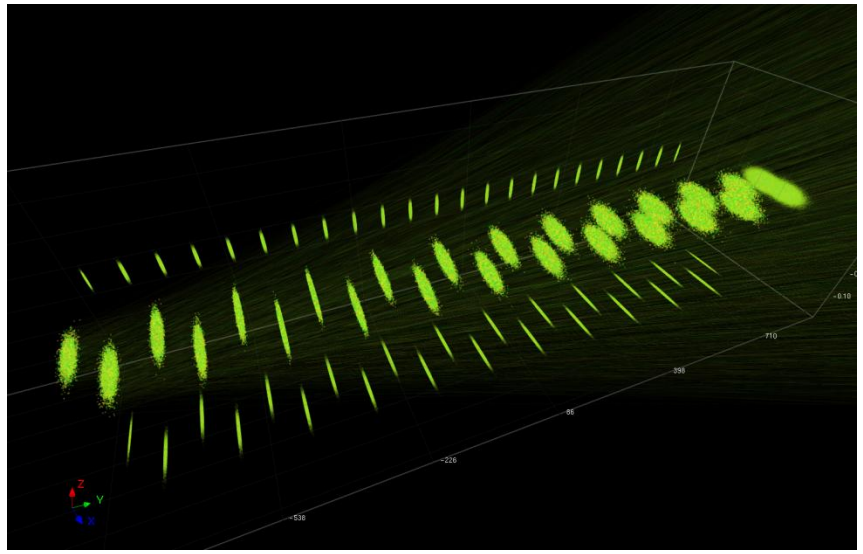
xrt is a free software library providing the tools for the X-ray propagation simulations – from fast and accurate physical models to plotting and visualization instruments.

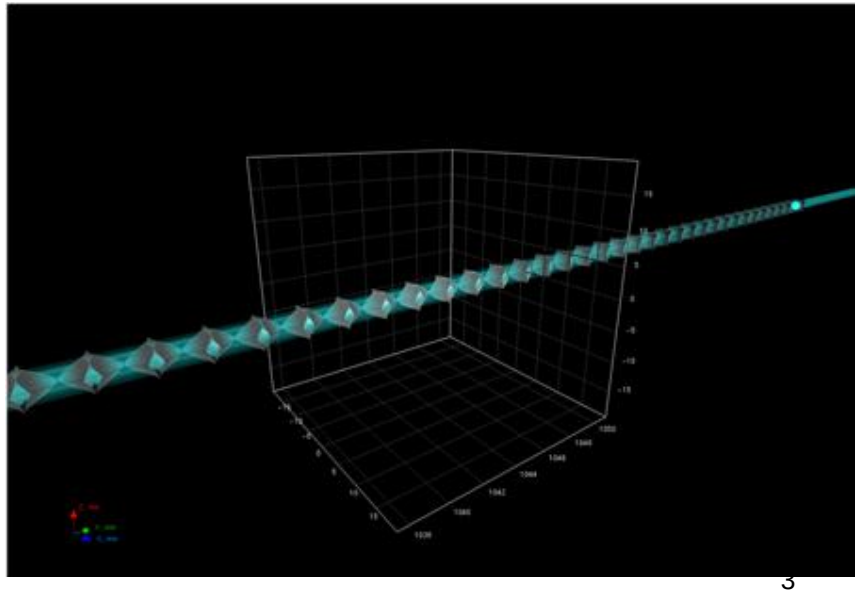
Monte-Carlo integration technique is used to calculate the beam flux and power density. xrt can handle millions of rays: more rays means better accuracy and higher quality plots.



Features

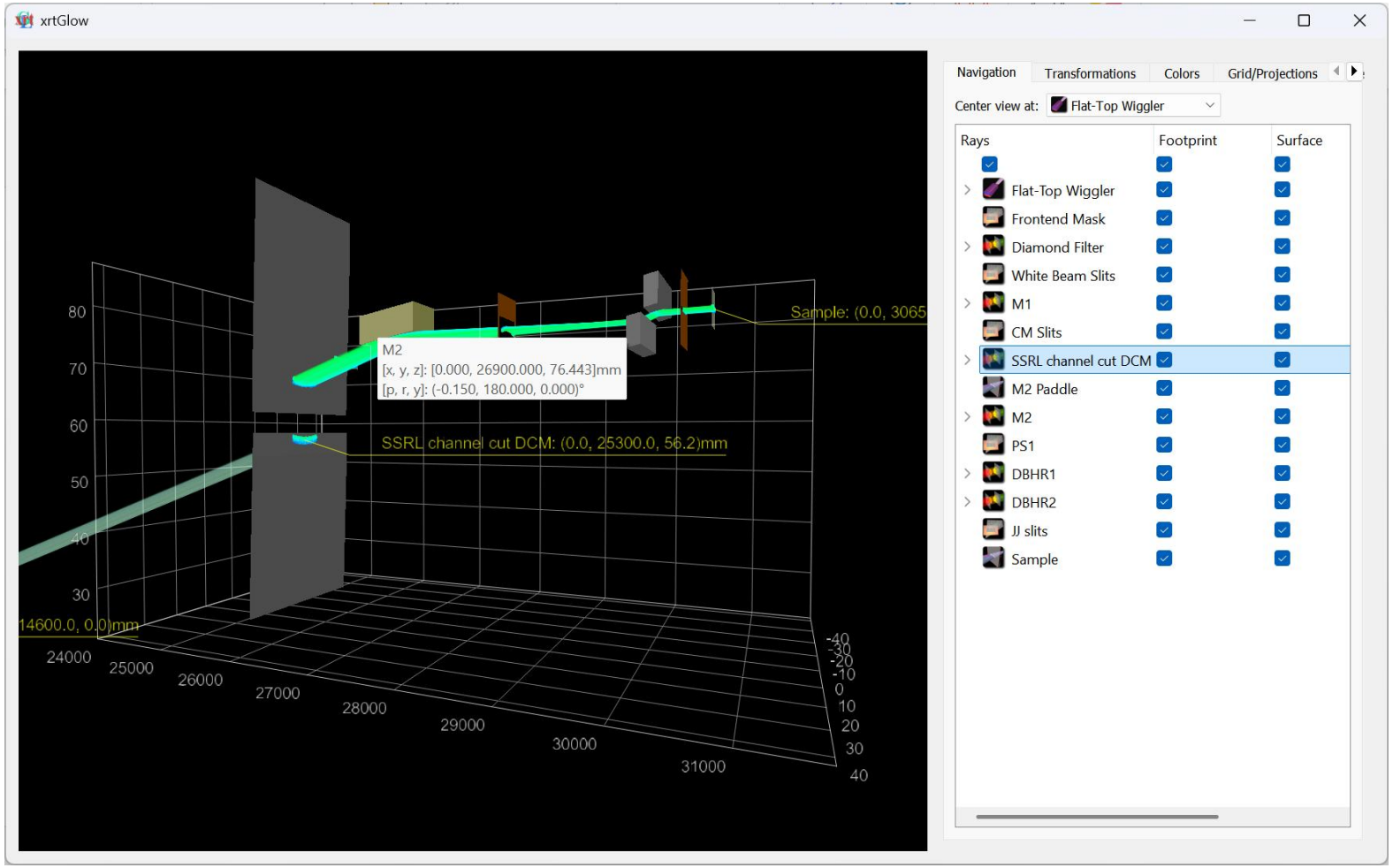
- Homogeneous environment: rays generation, propagation and plotting controlled by single python script.
- All optical elements defined in global coordinate system
- No intermediate file operations , no maps or interpolation: all properties calculated for each ray in real time
- GPGPU accelerated calculations (OpenCL)
- Embedded material properties (reflection/refraction/absorption). Now with bent crystals!
- Intensity in physical units from the source to the sample
- Propagation in rays, wave and hybrid mode
- GUI-aided script generation and 3D visualizer





xrtGlow – 3D Visualizer for beamlines

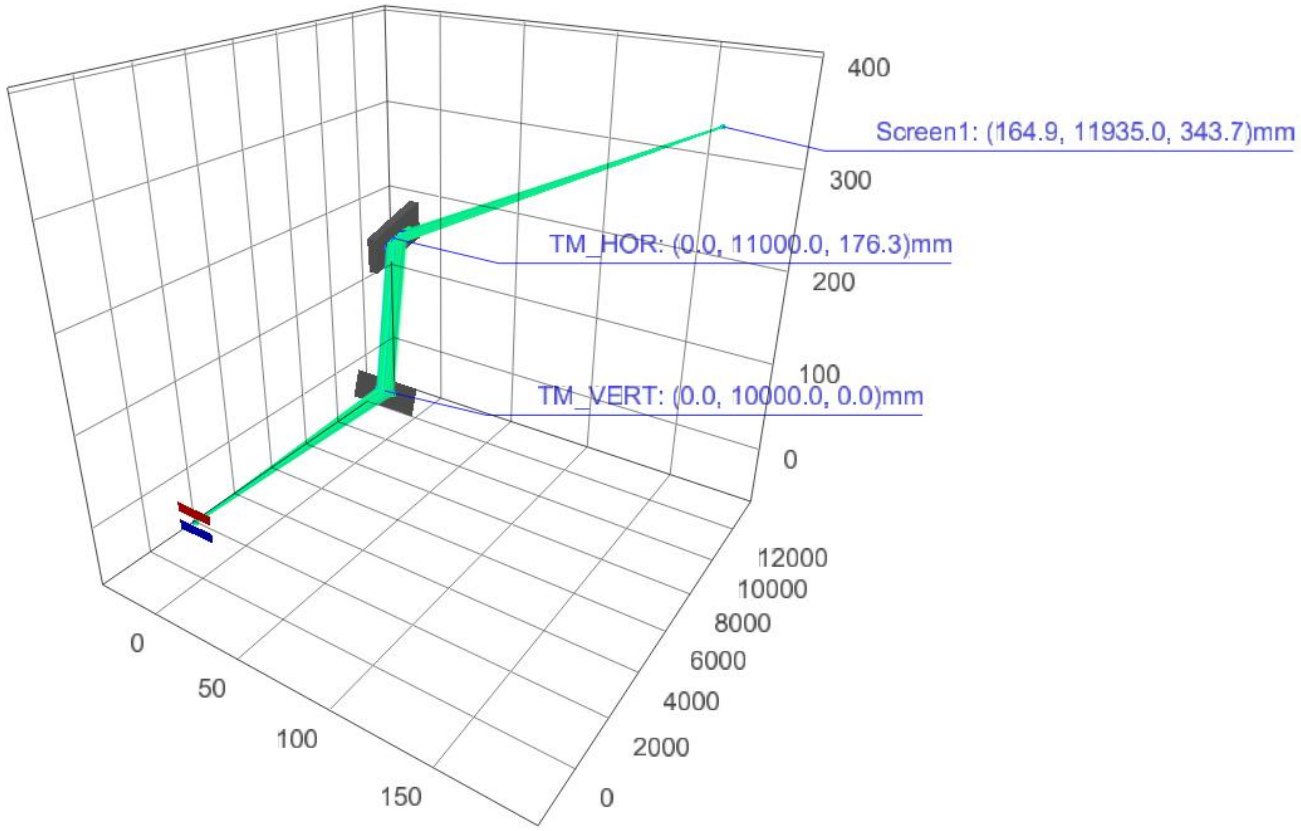
- ❑ Rendering in OpenGL, Qt UI (greatly improved in v2)
- ❑ Independently scalable coordinate axes
- ❑ Orthographic* and perspective projections
- ❑ Adjustable color axis
- ❑ Selection of the beam segments / footprints to plot
- ❑ Rendered optical elements
- ❑ Movable virtual screen *
- ❑ Transition between local and global coordinate systems
- ❑ **EPICS controls with pythonSoftIOC/asyncio**



** v2 implementation on the way*

Auto-generated PVs

```
def main():
    beamLine = build_beamline()
    beamLine.glow(v2=True, epicsPrefix='TST')
```



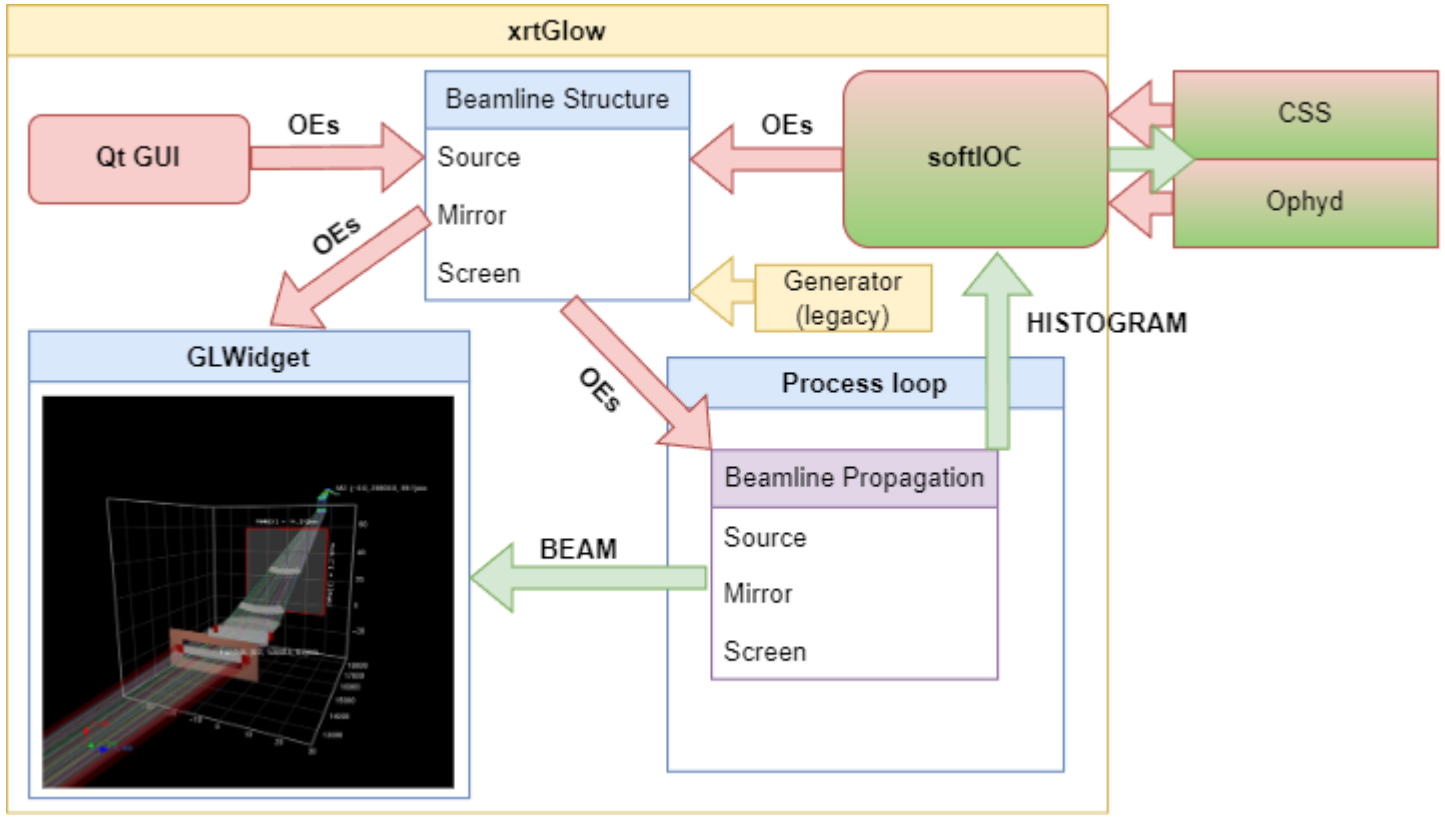
- Orientation
- Shape
- Data

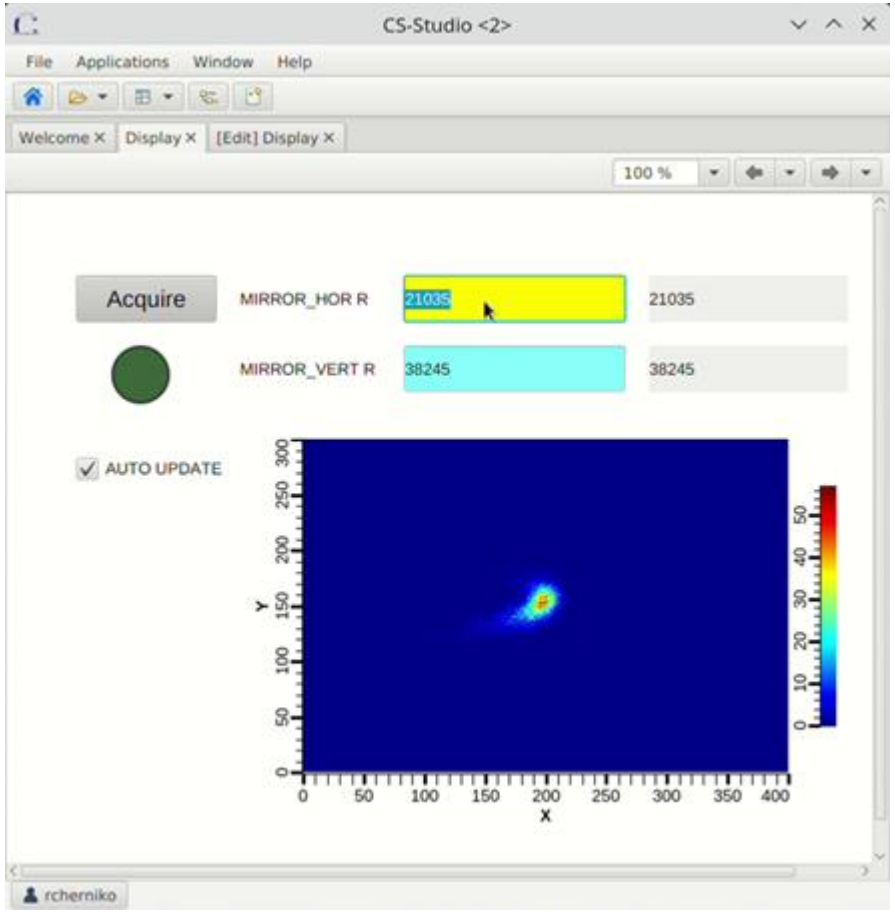
TST:TM_HOR:R
TST:TM_HOR:rotationSequence
TST:TM_HOR:r
TST:TM_HOR:limPhysX:lmin
TST:TM_HOR:limPhysX:lmax
TST:TM_HOR:yaw
TST:TM_HOR:limPhysY:lmin
TST:TM_HOR:limPhysY:lmax
TST:TM_HOR:roll
TST:TM_HOR:center:x
TST:TM_HOR:center:y
TST:TM_HOR:center:z
TST:TM_HOR:pitch

TST:Screen1:image
TST:Screen1:histShape:width
TST:Screen1:histShape:height
TST:Screen1:limPhysX:lmin
TST:Screen1:limPhysX:lmax
TST:Screen1:limPhysY:lmin
TST:Screen1:limPhysY:lmax
TST:Screen1:center:x
TST:Screen1:center:y
TST:Screen1:center:z

EPICS communication layer

Dynamic update workflow





Bluesky and blop for control and optimization

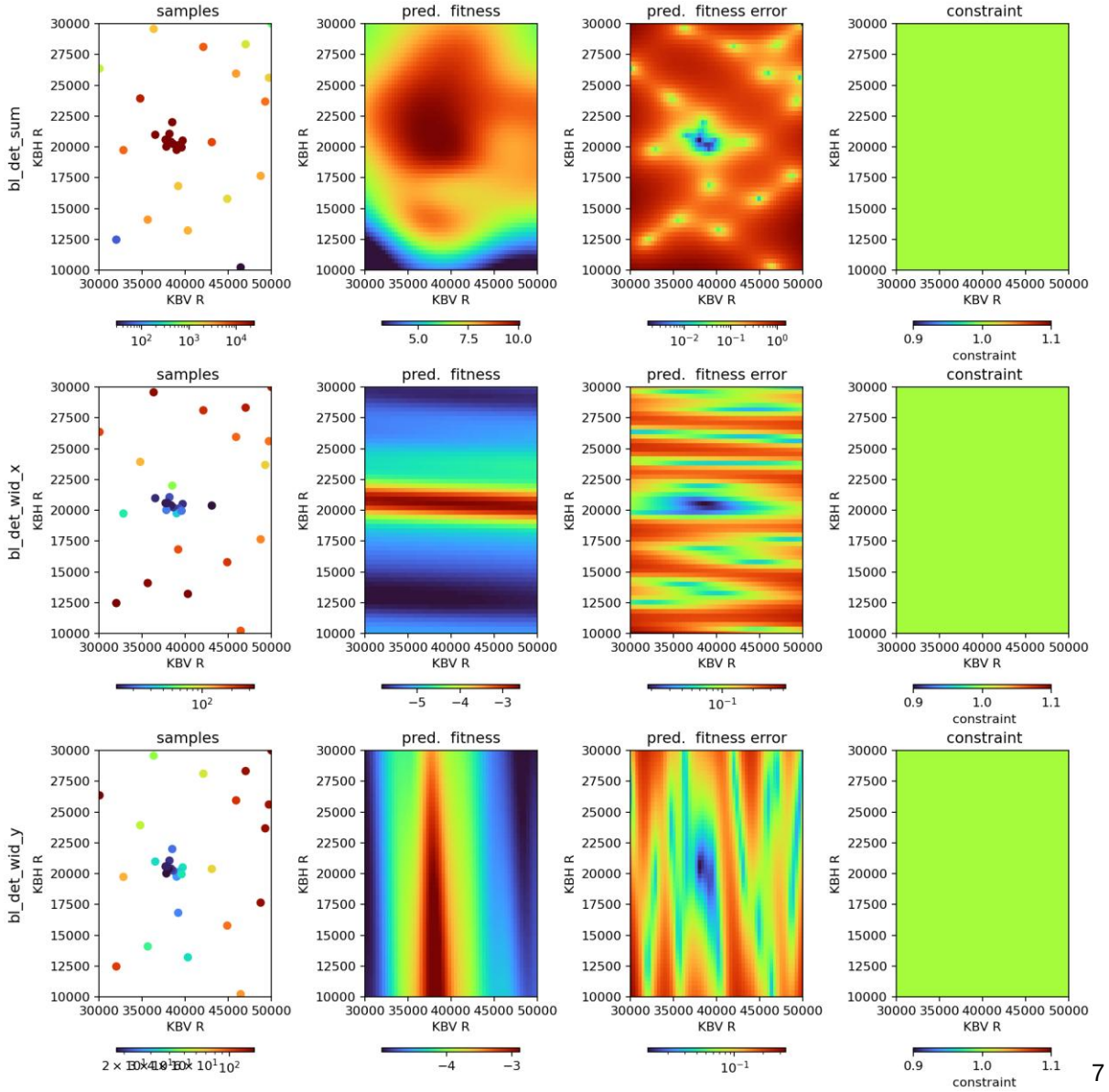
```python
class xrtEpicsScreen(Device):
    sum = Cpt(Signal, kind="hinted")
    max = Cpt(Signal, kind="normal")
    area = Cpt(Signal, kind="normal")
    cen_x = Cpt(Signal, kind="hinted")
    cen_y = Cpt(Signal, kind="hinted")
    wid_x = Cpt(Signal, kind="hinted")
    wid_y = Cpt(Signal, kind="hinted")
    image = Cpt(EpicsSignal, f'{epicsPrefix}:Screen1:image', kind="normal")
    acquire = Cpt(EpicsSignal, f'{epicsPrefix}:Acquire', kind="normal")
    acquireStatus = Cpt(EpicsSignal, f'{epicsPrefix}:AcquireStatus', kind="normal")
    image_shape = Cpt(Signal, value=(300, 400), kind="normal")
    noise = Cpt(Signal, kind="normal")

    def __init__(self, root_dir: str = "/tmp/blop/sim", verbose: bool = True,
                 noise: bool = True, *args, **kwargs):
        _ = make_dir_tree(datetime.now().year, base_path=root_dir)

        self._root_dir = root_dir
        self._verbose = verbose
class BeamlineEpics(Device):
    det = Cpt(xrtEpicsScreen, name="DetectorScreen")
    autoUpdate = Cpt(EpicsSignal, ':AutoUpdate', kind="normal")

    kbh_dsh = Cpt(EpicsSignal, ':TM_HOR:R', kind="hinted")
    kbv_dsv = Cpt(EpicsSignal, ':TM_VERT:R', kind="hinted")


    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
```

blop-xrt-examples : python — Konsole <2>

File Edit View Bookmarks Plugins Settings Help

New Tab Split View Copy Paste Find

4	15:38:50.8	36108	28680	2762.000	202.287	146.354	282.293	50.061
5	15:38:51.2	32848	29286	1741.000	191.500	146.602	255.300	92.429
6	15:38:51.5	31650	25875	2585.000	216.019	136.589	140.422	111.966
7	15:38:51.9	30258	23859	3657.000	202.461	140.740	114.540	164.763
8	15:38:52.2	32057	22360	11525.000	203.636	144.505	72.846	115.444
9	15:38:52.6	33917	20981	8035.500	194.664	142.625	14.012	66.538
10	15:38:52.9	32365	18805	11309.000	207.871	147.777	82.466	102.699
11	15:38:53.2	30574	17493	2075.000	208.446	150.819	158.458	144.595
12	15:38:53.6	33684	16044	3095.000	196.700	142.920	235.800	71.040
13	15:38:54.0	35426	16611	8121.500	204.774	147.416	222.627	50.190
14	15:38:54.3	34490	14300	1708.000	213.569	148.798	347.438	66.171
15	15:38:54.7	30631	10850	6.000	151.000	142.000	163.000	81.000
16	15:38:55.0	32531	11903	91.000	156.513	151.842	252.075	60.417
17	15:38:55.4	36496	13655	4274.000	207.810	146.426	325.481	31.715
18	15:38:55.7	37679	11495	-194.000	187.250	149.006	353.000	21.655
19	15:38:56.1	39635	14722	1621.000	215.833	148.336	288.767	40.216
20	15:38:56.4	40549	15025	5684.000	212.401	146.830	287.424	39.359
21	15:38:56.8	41355	14533	3094.000	211.886	146.786	319.971	48.460
22	15:38:57.1	42516	10620	104.000	181.008	148.500	341.917	43.900
23	15:38:57.5	47664	10983	14.000	107.425	137.000	214.850	53.400
24	15:38:57.8	49611	12794	54.000	199.217	151.075	398.433	130.450
25	15:38:58.2	45813	12357	75.000	188.938	154.000	377.875	113.500
26	15:38:58.5	44466	13182	285.000	195.875	169.875	313.250	70.750



Acknowledgements and resources

Team:

Konstantin Klementiev (MAX IV) - xrt
Max Rakitin - bluesky, blop
Thomas Morris - blop
Jennefer Maldonado - bluesky, blop
Thomas Hopkins - bluesky, blop
Jessica Moylan - blop

Code:

https://github.com/kklmn/xrt/tree/new_glow
https://github.com/bluesky/bluesky
https://github.com/NSLS-II/blop
https://github.com/yxrmz/blop-xrt-examples
https://github.com/yxrmz/profile_collection_xrt