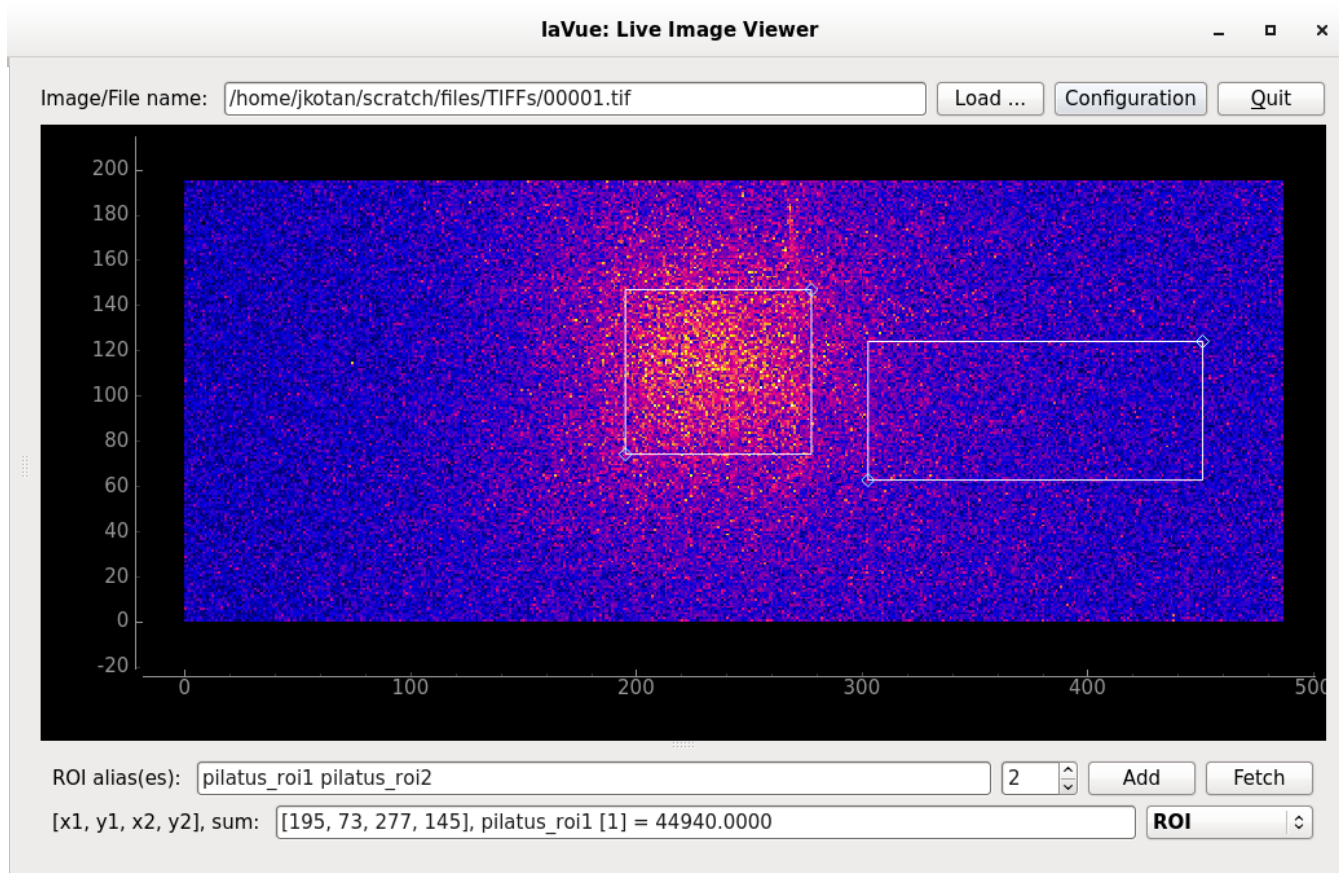


# LaVue - ROI Tool

ROI Tool selects Regions Of Interest and calculates a sum of their pixel intensities



- **ROI alias(es)** for roi tango devices
- **Add** to or **Fetch** from sardana environment (see below). If lavue is working with ANALYSISDEVICE , i.e. `lavue -n <device>`, ROIs bounds are sending also to tango SPECTRUM **Rols** attribute of the device, e.g. LambdaOnlineAnalysis server.
- **Sum** of the selected **ROI** or all **ROIs**. The used version can be selected in the configuration.

Moreover, in configuration can be set if the sums of calculated ROIs should be sent to **LavueController** tango server.

After adding ROIs to sardana the following environment variables are created or updated

- **DetectorROIs**: **JSON** dictionary with all Regions Of Interests ranges, e.g.  
{ "pilatus\_roi1": [[195, 73, 277, 145]], "pilatus\_roi2": [[305, 65, 455, 125]], "old\_pilatus\_roi": [[19, 27, 73, 146]] }
- **DetectorROIsValues**: **JSON** dictionary with Regions Of Interests sums, e.g.  
{ "pilatus\_roi1": [44940.0], "pilatus\_roi2": [8167.0] }
- **DetectorROIsParams**: **JSON** list of image transformations performed by lavue, e.g.  
["transpose", "flip-left-right", "flip-up-down"]
- **DetectorROIsOrder**: **JSON** list of ROI aliases representing they order, e.g.  
["pilatus\_roi1", "pilatus\_roi2"]

The **configuration** of the tool can be set with a **JSON** dictionary passed in the `--tool-configuration` option in command line or a `toolconfig` variable of `LavueController.LavueState` with the following keys:

`aliases` (list of strings), `rois_number` (integer)

e.g.

```
lavue -u roi --tool-configuration \{"rois_number":2,\ "aliases":["pilatus_roi1","\ "polatus_roi2"]}
```