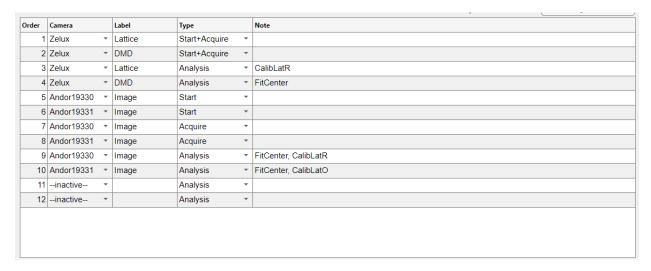
# Instruction for Setting Up SequenceTable

#### **Basic information**

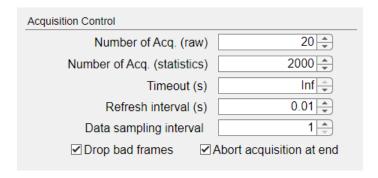
The acquisition and analysis pipeline are mostly defined as a SequenceTable.

An example of the table:



Additional to the SeuqenceTable, there are a few parameters to control the acquisition behaviors.

An example of the parameters:



## **Configure Sequence Table**

A step is a single row in the SequenceTable. To define a step, one need to specify 4 varaibles:

- Camera: categorical variable, the camera/projector to use for this step, can be selected from cameras {Andor19330, Andor19331, Zelux} or projectors {DMD}
- Label: text string, the label of the image, to distinguish two images taken by the same camera
  - Format is '<name>\_<wavelength>', where name cannot be blank string "". If not specified, wavelength is assumed to be 852 (nm). Example is 'Image', 'Pattern\_532', 'Lattice\_935'
- Type: categorical variable, type of operations, can be set to {Start, Start+Acquire, Acquire, Analysis, Project}
- **Note:** text string, optional, additional input parameters for the step, separated by comma, leaving empty means default settings

#### To define **Type**:

- "Start" mode starts the acquisition
  - o It either triggers the camera internally, or set the camera to be ready for external trigger
- "Acquire" mode consists of two processes: acquisition and preprocess
  - In acquisition, it acquires a raw image from specified camera with specified label
  - Immediately after acquisition it pre-processes the raw image
- "Start+Acquire" mode does both "Start" and "Acquire"
- "Analysis" mode performs a series of analysis processes
- "Project" mode only works for projectors

Note specifies the parameters for the step, which will be different depending on the Type:

- "Start", the note parameters control the behavior of starting acquisition.
  - o Available parameters are defined in class Camera.startAcqusition under /core/camera/Camera.m
- "Acquire"/"Start+Acquire", because it consists of more than one process, the parameters require an identifier to specify which process they are controlling
  - For "Acquire", available identifiers are {Acquire, Preprocess}
    - "Acquire", available parameters are defined in class Camera.acquire under /core/camera/Camera.m
    - "Preprocess", available parameters are defined in class Preprocessor.process under /core/preprocess/Preprocessor.m
  - o For "Start+Acquire", available identifiers are {Start, Acquire, Preprocess}
  - An example note is: "Start, verbose=1, Acquire, min\_wait=1, Preprocess, verbose=1"
- "Analysis", the parameters need to specify which analysis process to make as identifier and followed by the parameters
  - The available analysis processes are defined in class AnalysisRegistry under /core/analysis/AnalysisRegistry.m
- "Project", the parameters control the projected pattern

### **Configure Acquisition Parameters**

- **Number of Acq. (images):** control the storage space for raw images, the raw images generated by this number of full sequence run will be kept in memory.
- Number of Acq. (Statistics): control the storage space for analysis results, because the analysis usually gives only a few numbers, this number can be much larger than "Number of Acq. (images)" while still taking much smaller storage space.
- Timeout (seconds): control the maximum wait time for acquiring a single image from any camera. If wait time
  exceeds the max but there is still no new image available, the program will throw a warning and return an empty
  image.
- **Refresh interval (seconds)**: control the refresh interval for acquiring a single image from any camera. The camera will check if a new image is available every this interval and return the acquired image when it sees the new data.
- **Data sampling interval**: integer, control the saving interval of data acquisition. The app will save the data every this interval. When set to 1, it saves all the data.