# 1 – installation / first start

Browser: github, pypi

CMD: pip install dataArtist

**Folder**: images + numpy array + multiple plots

**Scope**: #######

1. Welcome / introduction (open source, imgProcessing, EL)
2. Where to find (github, pip)
3. How to install Win64-> portable // pip install
4. First start: portable: dataArtist.exe // cmd dataArtist
5. Help file, shortcuts, support(report bugs)
6. Drag/Drop TIFF, raw, numpy array
7. Dock management (tabbing, moving, closing, min/maximizing, fullscreen, context menu, pop out/in)
8. Moving and zooming
9. Colorbar: changing range, color scheme
10. Tools:
    1. Where to find (Toolbars …)
    2. Hide/show via menu or right click remove
    3. VIEW: hide axes, colorbar, rotate/transpose,
    4. MEASURE: get px values, measure
    5. LINK view and colorbar
11. Preferences side bar
    1. changing tabs, changes display
    2. change axes names
    3. view layer details
    4. set title
    5. title from layer
    6. limit layers (only explain, why)
12. Console output

# 2 – structure

Browser: github, pypi

CMD: pip install dataArtist

**Folder**: images + raw images + numpy array

**Scope**: understanding how dA is build: WS, Layers, states, sessions, tools, toolbars,

1. Load RAW
2. Loat set but crop (faster, less ram)
3. Current WS full, so create new workspace (first shortcut, than using menubar)
   1. Explain idea, that basically multiple windows in on
   2. Move one display from WS1 to WS2 (and move back to have WS free)
4. Layer:
   1. Load set of images (tiff) in one display
   2. Explain file import (together, separated, load files, show dialog)
   3. Time slider - arrow keys
   4. Tool: grid view (go back to normal view afterwards)
   5. Copy layer to new display
   6. Move (unfitting) layer to this display: NEED SAME SHAPE, or resize/warping…
5. Tool: line plot (on multi layer)
   1. Horiz/vert averaging
   2. No averaging (get snipped)
6. Toolbar stack:
   1. Show diff. as colorlayer (explain colorlayer tool)
   2. Slice
   3. Drill
7. Intro: while we are working current state saved every 15 min, goto:
   1. Preferences
   2. Mention max. saved states and that wil be explained later
   3. View: change colorscheme, explain idea behind different profiles
   4. And here we have out import options again …
   5. Communication: demonstrate watch folder
      1. Setup show no dialog and import import display
      2. drag drop imgs in folder
   6. Explain RabbitMQ, got website to show shat to download
      1. Explain commands, e.g. emitted by a labview session
8. Save current state (shortcut and menubar)
   1. Change something and save again
   2. Restore last state, show rename state, and that max n states to be found in prefs
   3. Show saved da file / double click on it to open it in new window
   4. Also drag/drop it to open it again

# 3 – measuring features

**Folder**: images

**scope**: scaling, measuring crack length, SNR, filter and save to file

1. Crosshair
2. Ruler
   1. Scale using length of cell
   2. Add unit using side bar
3. Selection tool:
   1. measure area of features
4. SNR:
   1. Do both (IEC first)
   2. My method (explain benefit of spatial resolved snr) – also result is more precise but slower
   3. At the end:
      1. we might want to make some notes: dragNdrop into dA to create notepad
      2. Also table possible
5. Save SNR to file

# 4 – automation

**Folder**: same as in 3

**scope**: access image data directly, commit simple image processing, automate executions of tools, create/load/save own script

UNDO /REDO

1. Subtract images
2. Import modules
3. Add tools
4. Show 2 examples

# 5 – camera calibration I

**Folder**: images: different focus, noise[multiple EL img pairs],dark current

**scope**: EL imaging (but also interesting for PL and normal img) get best focus, in order to correct images: calibrate lens, dark current – do all tools in toolbar from left to right

1. When toolbar calibrate opened: new [global] tool: ‘CalibrationFile’
   1. Explain purpose and create new file
2. Relative sharpness
   1. Load diff sharpness img (not all – leave 3 for later)
   2. Explain avail. Of different methods – propose tenengrad
   3. Execute on all layers
3. NLF
   1. Noise not constant over intensity
      1. SignalToNoiseRatio article// show fig 5, eq 11
   2. Is measurable using multiple pairs (of equivalent img)
   3. Load pairs – show individual differences (zoom)
   4. Exec tool
   5. Update calibration
   6. Show place in calibration
4. Dark current
   1. Describe bg removal with 1 EL and 1 BG img (subtract) -> hot px gone, but noise slightly increased (noise-noise), and extra bg img needed every time
   2. In case uniform light (dark chamber) bg img can be calc. from multiple bg img
      1. No need for extra img, lower noise level (show in comparison later)
   3. Load imgs
   4. 2 methods: only as function can be used to update camCal
   5. Exp. Times need to be given in layer value
   6. Explain idea of stack dimension [n] or n imgs or time …
   7. Get layer values from ‘individual’ or ‘fromName’
   8. Execute tool
   9. Update calFile
   10. Explain output
       1. Calculate bg img for on of the bg exp times
       2. compare result with normal bg img – put both in same display do line cut
          1. Same result but less noise

# 6 – camera calibration II

**Images**:

vignetting[calibr, and from EL imgs]+fitting bg img,

PSF,

lens

**Browser**: <http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html>

<http://scikit-image.org/docs/dev/api/skimage.restoration.html#skimage.restoration.unsupervised_wiener>

<https://en.wikipedia.org/wiki/Distortion_(optics)>

<http://www.ieee-pvsc.org/ePVSC/core_routines/view_abstract_no.php?show_close_window=yes&abstractno=673>

**Word**: CamCal article snipped on vign.)

**scope**: EL imaging (but also interesting for PL and normal img) vignetting, absolute sharpness via PSF, lens[barrel] distortion

1. Calibration based on PVSC proceeding (open browser
2. Flat field (=vignetting+ cam sens + dirt within system)
   * 1. Vignetting often biggest influence
   1. Load vignetting + bg imgs
      1. Explain how vign. Imgs done (show in word snipped)
   2. Execute tool and update calFile
   3. Show that bad cal. Imgs doesn’t influence output
   4. ALTERNATIVE when no flatField img there: vignetting ONLY
      1. Take normal imgs (same setup)
      2. All img averaged, smoothed and vignetting eq (kang-weiss)/polynomial fitted
      3. Do function and polynomial fit (compare to proper result)
3. Absolute sharpness
   1. Different methods to obtain PSF (e.g. slanted edge)
   2. Here only ‘mutliple pin holes’ supported at the moment
   3. How pattern is done
      1. Load corrected pattern img
      2. Explain needle, tape, smaller(<=50%) that px size
   4. Set boundaries, draw points
   5. Exec. Tool
   6. Show all different found pin holes
   7. Explain filter to remove low freq. noise
   8. update calFile
4. Deconvolution balance:
   1. Two different deconvolution filters avail:
      1. Wiener and unsupervised wiener (show browser – skimage)
      2. Wiener much faster, but needs extra balance value
      3. If that one not known imgCorr. Will use unsupervised wiener instead
   2. Loads 1 EL img
   3. Use ROI to create smaller subset (say is low on full img)
   4. Exec tool
   5. Scroll through results
   6. best looking img found – go back to orig. displ. And update calFile
5. Lens distortion (barrel distortion):
   1. Brower - wikipedia (often when wide angle/fish eye used)
   2. Needs to be corrected, even if not visible
      1. Demonstrate on EL image – disable. AspectRatio and skewing img
   3. Click on show patterns and explain different patterns avail
   4. Load set of chessb patt imgs
   5. Setup params (draw chessb. )
   6. Explain ‘’live’
   7. Exec tool
   8. Explain output and update calibration
6. calibrationFile tool
   1. setup light source and show different cal arrays

# 7– image correction

**Folder**: 3 EL sets (2EL (1 BG)) + devices at diff. positions + stitchable imgs

**scope**: how fast and easy it is to remove camera dependent and perspective image correction

1. CameraCorrection (save all output)
   1. Cryst module (2el)
   2. Same with sharpening (1 el)
   3. Low qual thinfilm (2 el 1 bg)
2. Perspective correction
   1. crystMod: auto detect quad – no param changes
   2. same – manual detect + intensity correction + show tilt factor map
   3. clean WS
   4. multiple img of on device in different pos.
      1. correct all using pattern recogn
   5. save all to file
3. img stitching
   1. ######

# 8 – modding

**Browser**: API

**Eclipse**: To show packages (open MedianFilterThreshold, GreatEyesImageFormat, moddingExample)

**scope**: make own tools, explain package structure

Tool example: MedianFilterThreshold

Show file, use and explain in dA

Reader example: GreatEyesImageFormat

Show file

Show ownGui.py