Numerical Methods in TEM

Christoph T. Koch

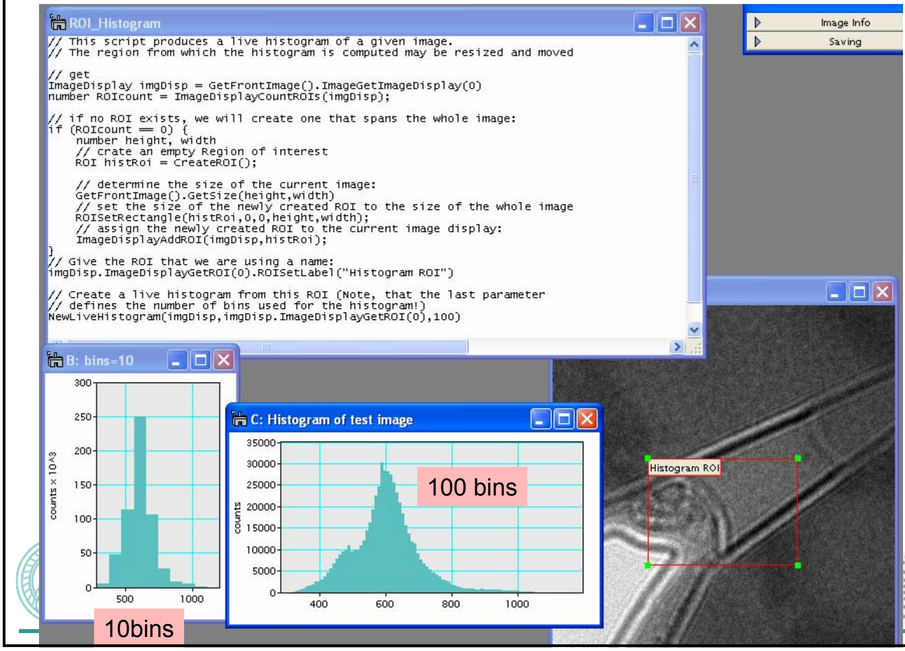
Max Planck Institut für Metallforschung

http://hrem.mpi-stuttgart.mpg.de/koch/Vorlesung

Outline of this lecture

- Computing Histograms
- Compute Diffractogram
- Follow up to last lecture on edge smoothing for FFTs
- More on Image Alignment
- Microscope Control using DM commands
- Calling any function that you find in the menu (and those that are not)

Computing a Histogram

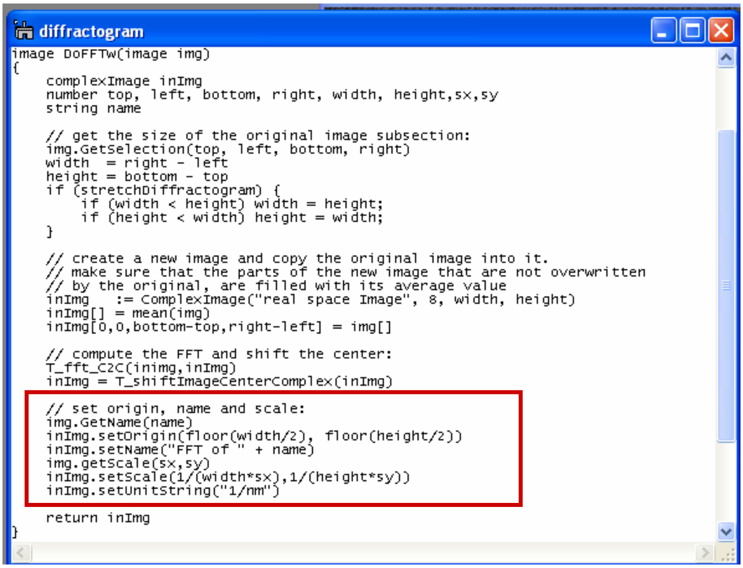


```
// This script produces a live histogram of a given image.
// The region from which the histogram is computed may be resized and moved

// get
ImageDisplay imgDisp = GetFrontImage().ImageGetImageDisplay(0)
number ROIcount = ImageDisplayCountROIs(imgDisp);

// if no ROI exists, we will create one that spans the whole image:
if (ROIcount == 0) {
    number height, width
    // crate an empty Region of interest
    ROI histRoi = CreateROI();

    // determine the size of the current image:
    GetFrontImage().GetSize(height,width)
    // set the size of the newly created ROI to the size of the whole image
    ROISetRectangle(histRoi,0,0,height,width);
    // assign the newly created ROI to the current image display:
    ImageDisplayAddROI(imgDisp,histRoi);
}
// Give the ROI that we are using a name:
imgDisp.ImageDisplayGetROI(0).ROISetLabel("Histogram ROI")

// Create a live histogram from this ROI (Note, that the last parameter
// defines the number of bins used for the histogram!)
NewLiveHistogram(imgDisp,imgDisp.ImageDisplayGetROI(0),100)
```

100 bins

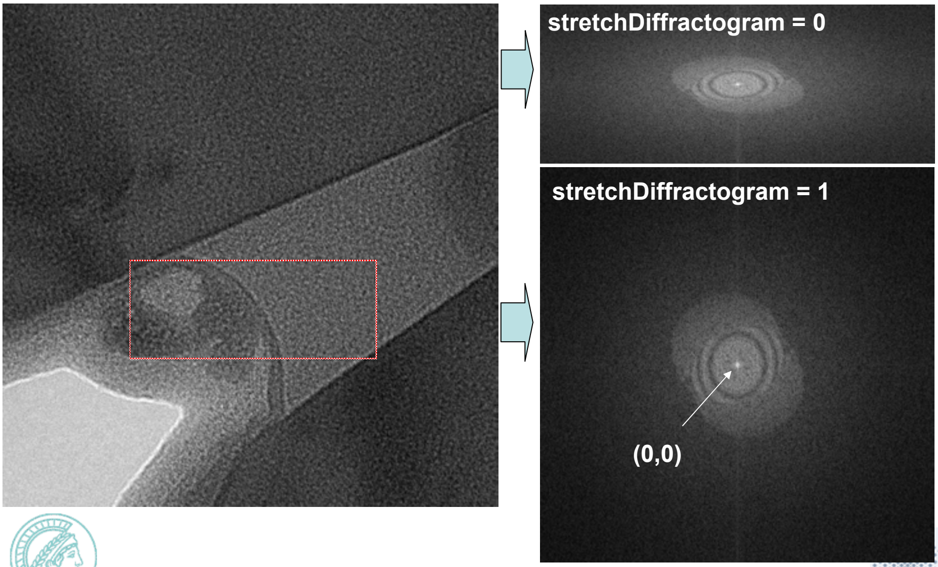
10bins

Compute Diffractograms of arbitrarily sized images

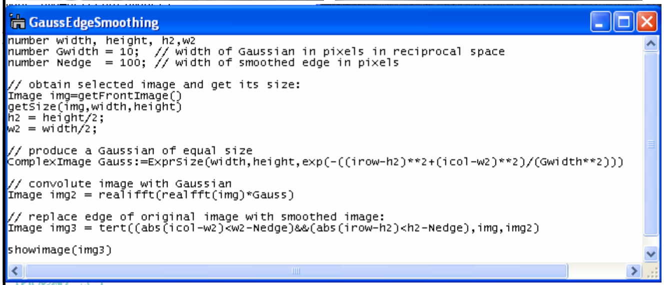


```
image DoFFTw(image img)
{
    complexImage inImg
    number top, left, bottom, right, width, height,sx,sy
    string name

    // get the size of the original image subsection:
    img.GetSelection(top, left, bottom, right)
    width  = right - left
    height = bottom - top
    if (stretchDiffractogram) {
        if (width < height) width = height;
        if (height < width) height = width;
    }

    // create a new image and copy the original image into it.
    // make sure that the parts of the new image that are not overwritten
    // by the original, are filled with its average value
    inImg    := ComplexImage("real space Image", 8, width, height)
    inImg[] = mean(img)
    inImg[0,0,bottom-top,right-left] = img[]

    // compute the FFT and shift the center:
    T_fft_C2C(inimg,inImg)
    inImg = T_shiftImageCenterComplex(inImg)

    // set origin, name and scale:
    img.GetName(name)
    inImg.setOrigin(floor(width/2), floor(height/2))
    inImg.setName("FFT of " + name)
    img.getScale(sx,sy)
    inImg.setScale(1/(width*sx),1/(height*sy))
    inImg.setUnitString("1/nm")

    return inImg
}
```

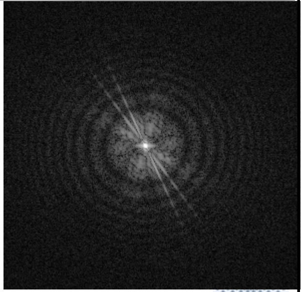
Diffractogram



Edge smoothing



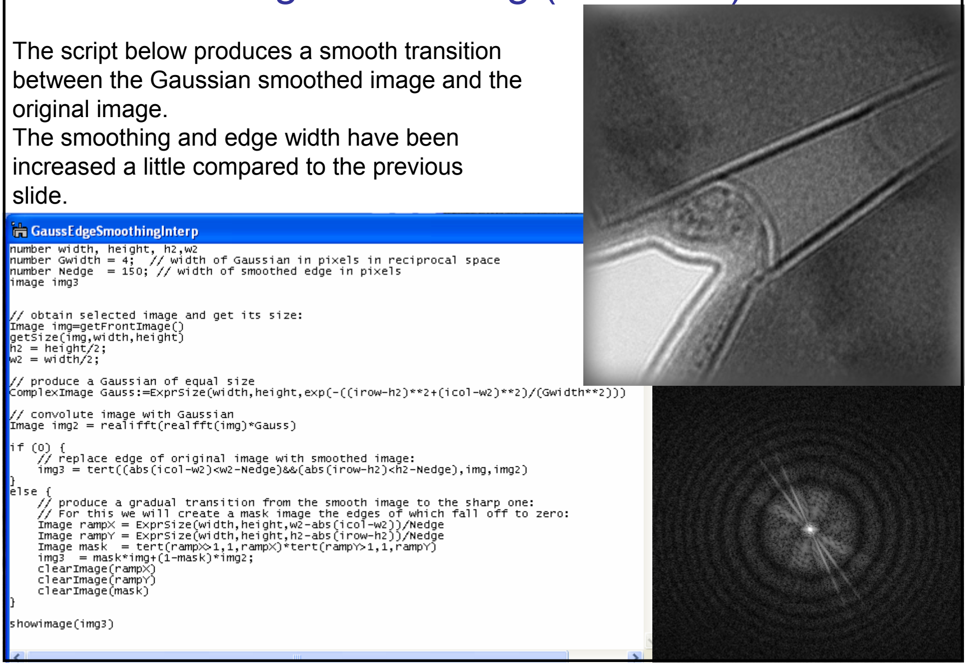
```
number width, height, h2,w2
number Gwidth = 10;  // width of Gaussian in pixels in reciprocal space
number Nedge  = 100; // width of smoothed edge in pixels

// obtain selected image and get its size:
Image img=getFrontImage()
getSize(img,width,height)
h2 = height/2;
w2 = width/2;

// produce a Gaussian of equal size
ComplexImage Gauss:=ExprSize(width,height,exp(-((irow-h2)**2+(icol-w2)**2)/(Gwidth**2)))

// convolute image with Gaussian
Image img2 = realifft(realfft(img)*Gauss)

// replace edge of original image with smoothed image:
Image img3 = tert((abs(icol-w2)<w2-Nedge)&&(abs(irow-h2)<h2-Nedge),img,img2)

showimage(img3)
```



Edge smoothing (version 2)

The script below produces a smooth transition between the Gaussian smoothed image and the original image.

The smoothing and edge width have been increased a little compared to the previous slide.



```
number width, height, h2,w2
number Gwidth = 4;  // width of Gaussian in pixels in reciprocal space
number Nedge  = 150; // width of smoothed edge in pixels
image img3


// obtain selected image and get its size:
Image img=getFrontImage()
getSize(img,width,height)
h2 = height/2;
w2 = width/2;

// produce a Gaussian of equal size
ComplexImage Gauss:=ExprSize(width,height,exp(-((irow-h2)**2+(icol-w2)**2)/(Gwidth**2)))

// convolute image with Gaussian
Image img2 = realifft(realfft(img)*Gauss)

if (0) {
    // replace edge of original image with smoothed image:
    img3 = tert((abs(icol-w2)<w2-Nedge)&&(abs(irow-h2)<h2-Nedge),img,img2)
}
else {
    // produce a gradual transition from the smooth image to the sharp one:
    // For this we will create a mask image the edges of which fall off to zero:
    Image rampX = ExprSize(width,height,w2-abs(icol-w2))/Nedge
    Image rampY = ExprSize(width,height,h2-abs(irow-h2))/Nedge
    Image mask  = tert(rampX>1,1,rampX)*tert(rampY>1,1,rampY)
    img3  = mask*img+(1-mask)*img2;
    clearImage(rampX)
    clearImage(rampY)
    clearImage(mask)
}

showimage(img3)
```

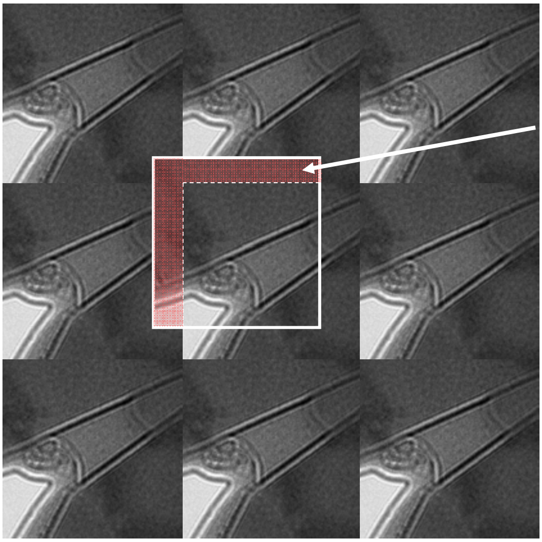
Image alignment: Cross correlation or χ^2 ?

$$\chi^2(\Delta\vec{r}) = \sum\left[I_1(\vec{r}) - I_2(\vec{r}+\Delta\vec{r})\right]^2$$

$$= \sum I_1{}^2 - \underbrace{2\sum I_1(\vec{r})I_2(\vec{r}+\Delta\vec{r})}_{\text{Cross correlation computes only this}} + \sum I_2{}^2$$

Cross correlation computes only this

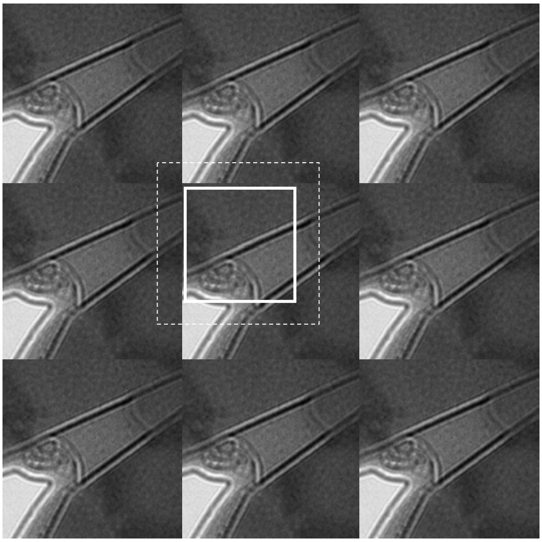
Cross correlation of whole images



Cross correlation also multiplies the images in the shaded region.

This will produce **errors**, if the overlap is non-negligible !!!

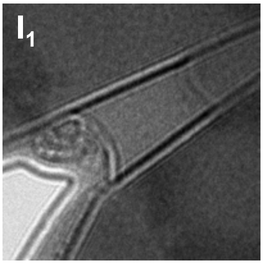
Solution: comparison of images of different size

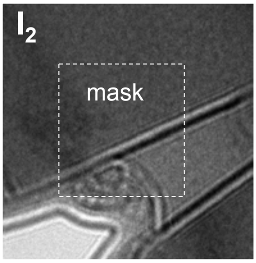


No overlap anymore!

Image alignment: χ^2–comparison

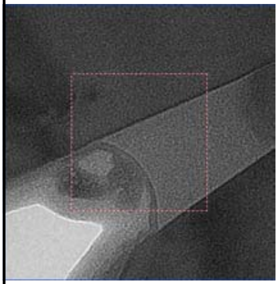
$$\begin{aligned}\chi^2(\Delta\vec{r}) &= \sum\left[I_1(mask) - I_2(mask+\Delta\vec{r})\right]^2 \\ &= I_1^{\,2} \otimes mask - 2I_1 \otimes \left(I_2 \cdot mask\right) + \sum I_2(mask)^2 \\ &= FT^{-1}\left\{FT\left[I_1^2\right]\cdot FT\left[mask\right]^* - 2FT\left[I_1\right]\cdot FT\left[I_2\cdot mask\right]^*\right\} + \sum\left(I_2\cdot mask\right)^2\end{aligned}$$



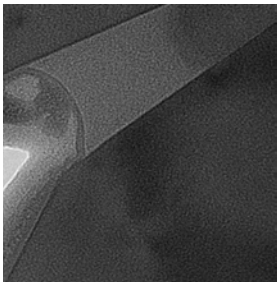


This requires 4 FFTs and 1 inverse FFT, but is much more reliable than simple cross correlation.

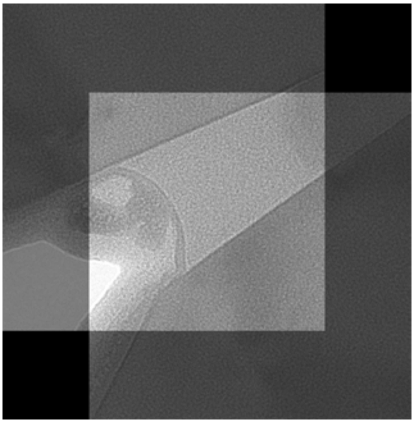
Adding 2 Images



+

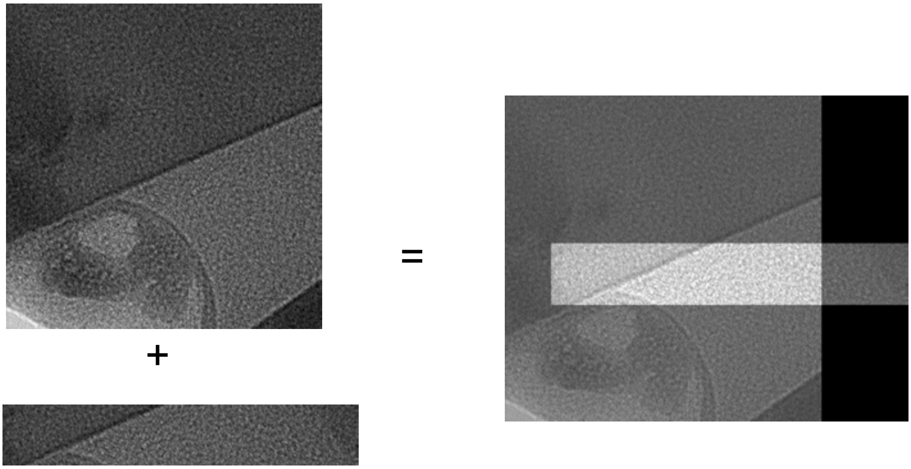


=



This image summation was done using the script ‘SelectAndAlign2Images’

Adding Images of different size



This image summation was done using the script
‘SelectAndAlign2Images’

Microscope Control Commands

DM provides a few general (independent of manufacturer) microscope control commands:

- EMSetupCommunication()
- EMCloseCommunication()
- EMPrepareTilt() … EMChangeTilt(int tx, int ty) // beam tilt
- EMPrepareStigmation() … EMChangeStigmation(int sx, int sy)
- EMPrepareShift() … EMBeamShift(int bx, int by)
- EMPrepareImageShift() … EMImageShift(int ix, int iy)
- EMChangeFocus(int df)

Note, that some of the above commands require prior initialization of the microscope communication (i.e. EMPrepare… - functions), so that subsequent commands are correctly interpeted.

A few Gatan Image Filter (GIF) Commands

- ifsetupcommunication()
- ifcgetSlitIn()
- ifcsetSlitIn(flag) // flag=0: retract, flag=1: insert

Submitting Commands to a JEOL microscope

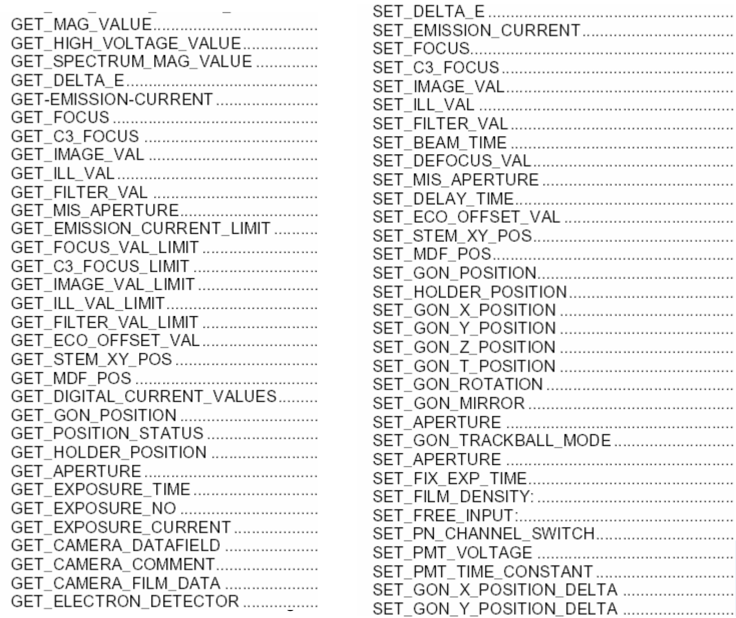
```
number pass
string command=""“

// execute an infinite loop which asks for a new command, send it to the
    microscope and quits, if the user hits the “Cancel” button.
while (1) {
    if(!getstring("Enter JEOL command string", command,  command)) {
        exit(0)
    }
    JEOLcommand(command,pass)
    beep()
}
```

Submitting Commands to a ZEISS microscope

```
string reply
string command=""

// execute an infinite loop which asks for a new command, send it to the
    microscope and quits, if the user hits the "Cancel" button.
while (1) {
    if(!getstring("Enter ZEISS command string", command,  command)) {
        exit(0)
    }
    LEO_command(command,reply)
    result("Microscope reply: "+reply+"\n")
}
```

The Zeiss Remote Command Protocol

.. From the table of content of the remote list of Zeiss [912 and **SESAM**] RS232 (serial port) commands ...
(see lecture website for complete list)



GET_MAG_VALUE..................................
GET_HIGH_VOLTAGE_VALUE................
GET_SPECTRUM_MAG_VALUE ....
GET_DELTA_E..
GET-EMISSION-CURRENT.......................
GET_FOCUS...
GET_C3_FOCUS ....
GET_IMAGE_VAL ....
GET_ILL_VAL..
GET_FILTER_VAL ....
GET_MIS_APERTURE................................
GET_EMISSION_CURRENT_LIMIT ....
GET_FOCUS_VAL_LIMIT ....
GET_C3_FOCUS_LIMIT ....
GET_IMAGE_VAL_LIMIT ....
GET_ILL_VAL_LIMIT.................................
GET_FILTER_VAL_LIMIT ....
GET_ECO_OFFSET_VAL.........................
GET_STEM_XY_POS ....
GET_MDF_POS ....
GET_DIGITAL_CURRENT_VALUES.........
GET_GON_POSITION ....
GET_POSITION_STATUS ....
GET_HOLDER_POSITION ....
GET_APERTURE......................................
GET_EXPOSURE_TIME ....
GET_EXPOSURE_NO ....
GET_EXPOSURE_CURRENT...................
GET_CAMERA_DATAFIELD ....
GET_CAMERA_COMMENT.......................
GET_CAMERA_FILM_DATA ....
GET_ELECTRON_DETECTOR ....

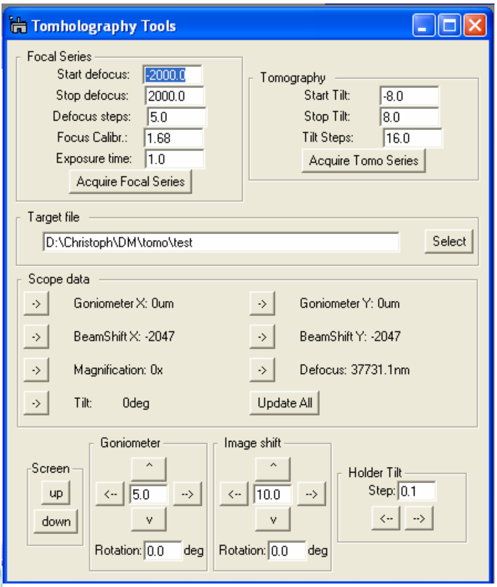
SET_DELTA_E ....
SET_EMISSION_CURRENT..................................
SET_FOCUS...
SET_C3_FOCUS ....
SET_IMAGE_VAL...
SET_ILL_VAL ....
SET_FILTER_VAL..
SET_BEAM_TIME ....
SET_DEFOCUS_VAL...
SET_MIS_APERTURE ....
SET_DELAY_TIME..
SET_ECO_OFFSET_VAL ....
SET_STEM_XY_POS..
SET_MDF_POS...
SET_GON_POSITION...
SET_HOLDER_POSITION.....................................
SET_GON_X_POSITION ....
SET_GON_Y_POSITION ....
SET_GON_Z_POSITION ....
SET_GON_T_POSITION ....
SET_GON_ROTATION ....
SET_GON_MIRROR ....
SET_APERTURE ....
SET_GON_TRACKBALL_MODE..........................
SET_APERTURE ....
SET_FIX_EXP_TIME...
SET_FILM_DENSITY: ....
SET_FREE_INPUT:...
SET_PN_CHANNEL_SWITCH................................
SET_PMT_VOLTAGE ....
SET_PMT_TIME_CONSTANT ....
SET_GON_X_POSITION_DELTA ....
SET_GON_Y_POSITION_DELTA ....

Examples for Zeiss commands

```
void LargeScreenUp() {
  string reply
  Leo_Command("T311",reply)
}


void LargeScreenDown() {
  string reply
  Leo_Command("T312",reply)
}


void Switch2DiffMode() {
  string reply
  Leo_Command("T002",reply)
}


void Switch2ImageMode() {
  string reply
  Leo_Command("T001",reply)
}
```

```
void Set_Image_Shift(number x,number y)  {
        string reply
        string command, commandx,commandy


        Leo_Command("S003",reply)  // set controls to Image Shift
        x = round(2047+x);
        y = round(2047+y);
        if (x<0)    {x=0}
        if (x>4095) {x=4095}
        if (y<0)    {y=0}
        if (y>4095) {y=4095}

        commandx = ""+x;
        commandy = ""+y;
        // produce a string with properly formatted numbers
        if      (x<10)   {commandx = "000"+x;}
           else if (x<100)  {commandx =  "00"+x;}
              else if (x<1000) {commandx =   "0"+x;}
        if      (y<10)   {commandy = "000"+y;}
           else if (y<100)  {commandy =  "00"+y;}
              else if (y<1000) {commandy =   "0"+y;}
        command = "S204"+commandx+commandy
        Leo_Command(command,reply)
}
```

Controlling the Zeiss SESAM



Tomholography Tools

Focal Series
Start defocus: -2000.0
Stop defocus: 2000.0
Defocus steps: 5.0
Focus Calibr.: 1.68
Exposure time: 1.0

Tomography
Start Tilt: -8.0
Stop Tilt: 8.0
Tilt Steps: 16.0

Target file
D:\Christoph\DM\tomo\test

Scope data
Goniometer X: 0um
Goniometer Y: 0um
BeamShift X: -2047
BeamShift Y: -2047
Magnification: 0x
Defocus: 37731.1nm
Tilt: 0deg

Goniometer: 5.0
Rotation: 0.0 deg

Image shift: 10.0
Rotation: 0.0 deg

Holder Tilt
Step: 0.1

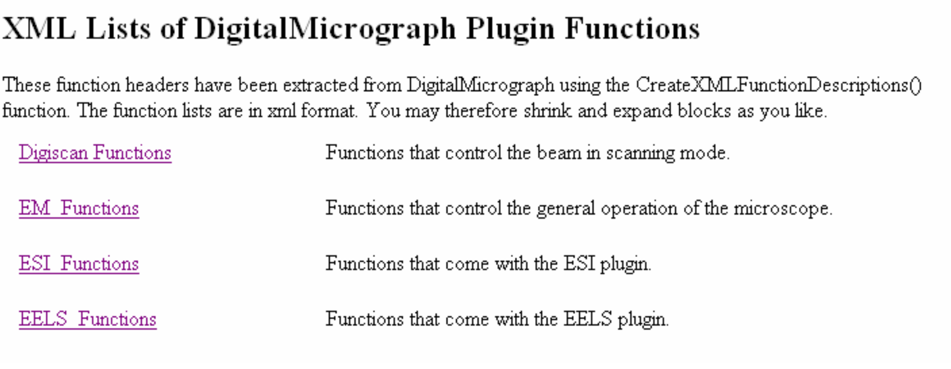
This is an example of a graphical user interface (GUI) which controls a number of microscope functions to perform a certain set of actions (e.g. acquire a tomographic series of focal series)

Calling commands from the DM-menu

- *Boolean ChooseMenuItem(String menu, String subMenu, String item)*
  - e.g. ChooseMenuItem(“EFTEMSI",“Artifact correction",“Remove X-rays")
  - e.g. ChooseMenuItem(“File”,””,”Print…”)

Undocumented DM functions

Digitalmicrograph contains a number of functions that are not officially documented anywhere. A few function names (and parameter lists) have been posted to this lecture’s website.



XML Lists of DigitalMicrograph Plugin Functions

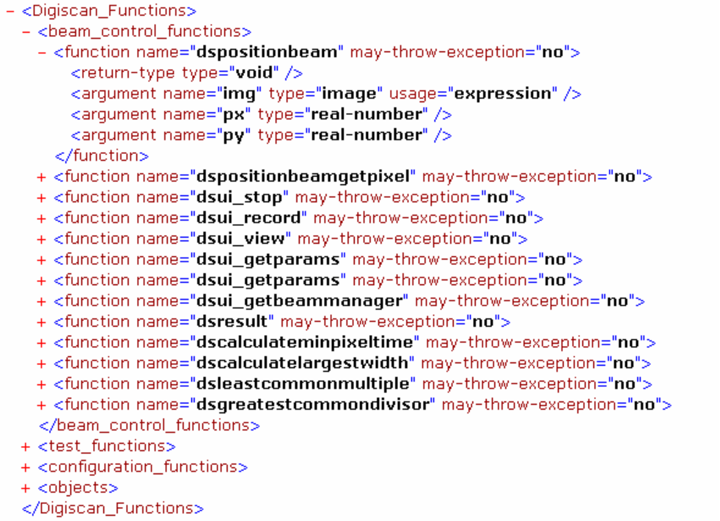
These function headers have been extracted from DigitalMicrograph using the CreateXMLFunctionDescriptions() function. The function lists are in xml format. You may therefore shrink and expand blocks as you like.

Digiscan Functions	Functions that control the beam in scanning mode.
EM Functions	Functions that control the general operation of the microscope.
ESI Functions	Functions that come with the ESI plugin.
EELS Functions	Functions that come with the EELS plugin.

http://hrem.mpi-stuttgart.mpg.de/koch/Vorlesung/Script/DM_PluginFunctions.html

Controlling Digiscan (VG / SESAM)

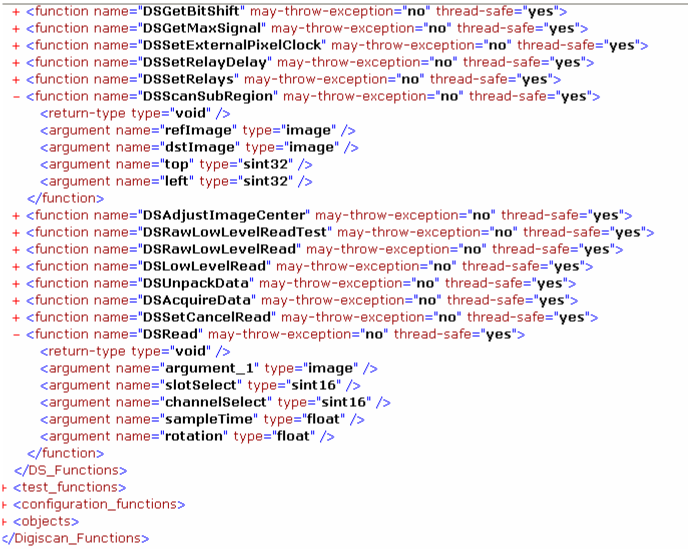
A list of DIGISCAN control commands which may be found online



```
- <Digiscan_Functions>
  - <beam_control_functions>
    - <function name="dspositionbeam" may-throw-exception="no">
        <return-type type="void" />
        <argument name="img" type="image" usage="expression" />
        <argument name="px" type="real-number" />
        <argument name="py" type="real-number" />
      </function>
    + <function name="dspositionbeamgetpixel" may-throw-exception="no">
    + <function name="dsui_stop" may-throw-exception="no">
    + <function name="dsui_record" may-throw-exception="no">
    + <function name="dsui_view" may-throw-exception="no">
    + <function name="dsui_getparams" may-throw-exception="no">
    + <function name="dsui_getparams" may-throw-exception="no">
    + <function name="dsui_getbeammanager" may-throw-exception="no">
    + <function name="dsresult" may-throw-exception="no">
    + <function name="dscalculateminpixeltime" may-throw-exception="no">
    + <function name="dscalculatelargestwidth" may-throw-exception="no">
    + <function name="dsleastcommonmultiple" may-throw-exception="no">
    + <function name="dsgreatestcommondivisor" may-throw-exception="no">
    </beam_control_functions>
  + <test_functions>
  + <configuration_functions>
  + <objects>
  </Digiscan_Functions>
```

http://hrem.mpi-stuttgart.mpg.de/koch/Vorlesung/Script/DM_PluginFunctions.html

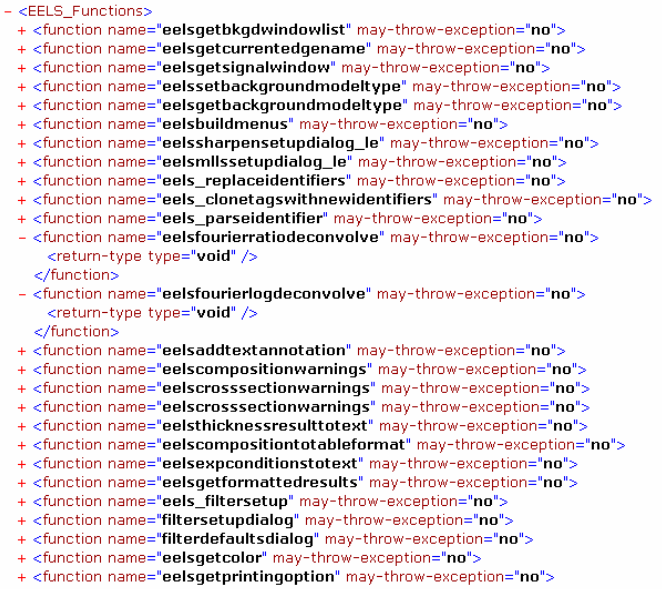
More DigiScan Commands



```
  + <function name="DSGetBitShift" may-throw-exception="no" thread-safe="yes">
  + <function name="DSGetMaxSignal" may-throw-exception="no" thread-safe="yes">
  + <function name="DSSetExternalPixelClock" may-throw-exception="no" thread-safe="yes">
  + <function name="DSSetRelayDelay" may-throw-exception="no" thread-safe="yes">
  + <function name="DSSetRelays" may-throw-exception="no" thread-safe="yes">
  - <function name="DSScanSubRegion" may-throw-exception="no" thread-safe="yes">
      <return-type type="void" />
      <argument name="refImage" type="image" />
      <argument name="dstImage" type="image" />
      <argument name="top" type="sint32" />
      <argument name="left" type="sint32" />
    </function>
  + <function name="DSAdjustImageCenter" may-throw-exception="no" thread-safe="yes">
  + <function name="DSRawLowLevelReadTest" may-throw-exception="no" thread-safe="yes">
  + <function name="DSRawLowLevelRead" may-throw-exception="no" thread-safe="yes">
  + <function name="DSLowLevelRead" may-throw-exception="no" thread-safe="yes">
  + <function name="DSUnpackData" may-throw-exception="no" thread-safe="yes">
  + <function name="DSAcquireData" may-throw-exception="no" thread-safe="yes">
  + <function name="DSSetCancelRead" may-throw-exception="no" thread-safe="yes">
  - <function name="DSRead" may-throw-exception="no" thread-safe="yes">
      <return-type type="void" />
      <argument name="argument_1" type="image" />
      <argument name="slotSelect" type="sint16" />
      <argument name="channelSelect" type="sint16" />
      <argument name="sampleTime" type="float" />
      <argument name="rotation" type="float" />
    </function>
  </DS_Functions>
+ <test_functions>
+ <configuration_functions>
+ <objects>
</Digiscan_Functions>
```

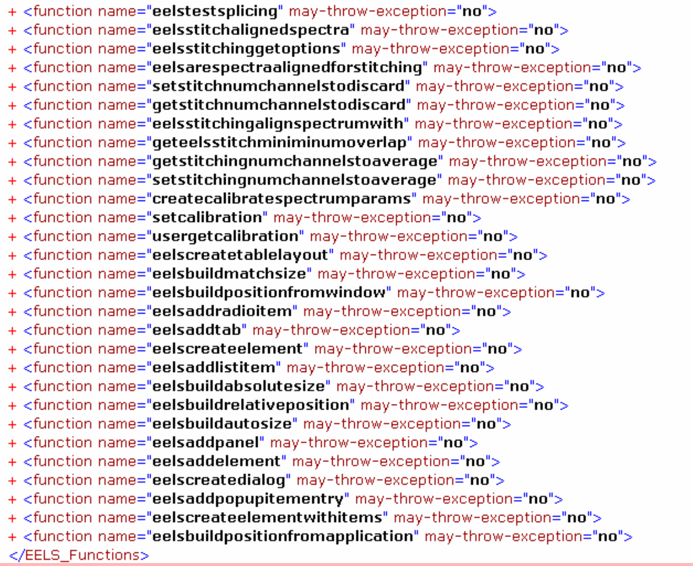
Scripts may acquire images, scan sub-regions, etc.

EELS Commands



```
- <EELS_Functions>
  + <function name="eelsgetbkgdwindowlist" may-throw-exception="no">
  + <function name="eelsgetcurrentedgename" may-throw-exception="no">
  + <function name="eelsgetsignalwindow" may-throw-exception="no">
  + <function name="eelssetbackgroundmodeltype" may-throw-exception="no">
  + <function name="eelsgetbackgroundmodeltype" may-throw-exception="no">
  + <function name="eelsbuildmenus" may-throw-exception="no">
  + <function name="eelssharpensetupdialog_le" may-throw-exception="no">
  + <function name="eelsmllssetupdialog_le" may-throw-exception="no">
  + <function name="eels_replaceidentifiers" may-throw-exception="no">
  + <function name="eels_clonetagswithnewidentifiers" may-throw-exception="no">
  + <function name="eels_parseidentifier" may-throw-exception="no">
  - <function name="eelsfourierratiodeconvolve" may-throw-exception="no">
      <return-type type="void" />
    </function>
  - <function name="eelsfourierlogdeconvolve" may-throw-exception="no">
      <return-type type="void" />
    </function>
  + <function name="eelsaddtextannotation" may-throw-exception="no">
  + <function name="eelscompositionwarnings" may-throw-exception="no">
  + <function name="eelscrosssectionwarnings" may-throw-exception="no">
  + <function name="eelscrosssectionwarnings" may-throw-exception="no">
  + <function name="eelsthicknessresulttotext" may-throw-exception="no">
  + <function name="eelscompositiontotableformat" may-throw-exception="no">
  + <function name="eelsexpconditionstotext" may-throw-exception="no">
  + <function name="eelsgetformattedresults" may-throw-exception="no">
  + <function name="eels_filtersetup" may-throw-exception="no">
  + <function name="filtersetupdialog" may-throw-exception="no">
  + <function name="filterdefaultsdialog" may-throw-exception="no">
  + <function name="eelsgetcolor" may-throw-exception="no">
  + <function name="eelsgetprintingoption" may-throw-exception="no">
```

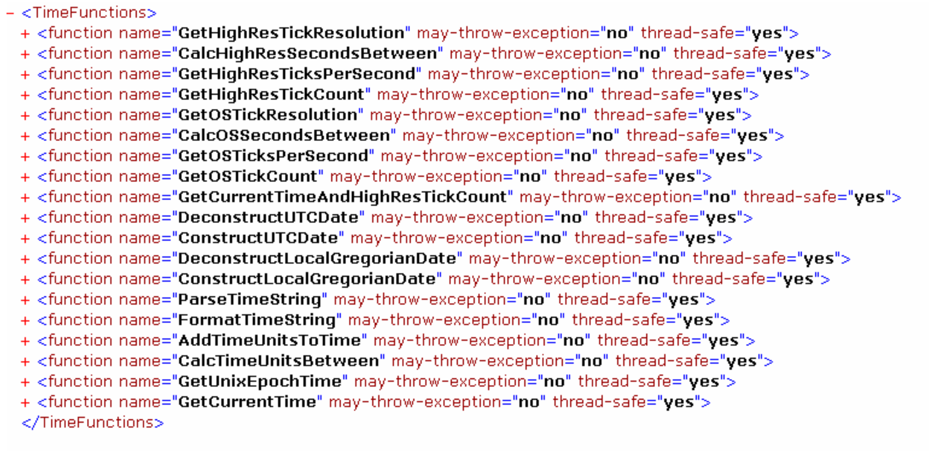
EELS Commands (2)



```
  + <function name="eelstestsplicing" may-throw-exception="no">
  + <function name="eelsstitchalignedspectra" may-throw-exception="no">
  + <function name="eelsstitchinggetoptions" may-throw-exception="no">
  + <function name="eelsarespectraalignedforstitching" may-throw-exception="no">
  + <function name="setstitchnumchannelstodiscard" may-throw-exception="no">
  + <function name="getstitchnumchannelstodiscard" may-throw-exception="no">
  + <function name="eelsstitchingalignspectrumwith" may-throw-exception="no">
  + <function name="geteelsstitchminiminumoverlap" may-throw-exception="no">
  + <function name="getstitchingnumchannelstoaverage" may-throw-exception="no">
  + <function name="setstitchingnumchannelstoaverage" may-throw-exception="no">
  + <function name="createcalibratespectrumparams" may-throw-exception="no">
  + <function name="setcalibration" may-throw-exception="no">
  + <function name="usergetcalibration" may-throw-exception="no">
  + <function name="eelscreatetablelayout" may-throw-exception="no">
  + <function name="eelsbuildmatchsize" may-throw-exception="no">
  + <function name="eelsbuildpositionfromwindow" may-throw-exception="no">
  + <function name="eelsaddradioitem" may-throw-exception="no">
  + <function name="eelsaddtab" may-throw-exception="no">
  + <function name="eelscreateelement" may-throw-exception="no">
  + <function name="eelsaddlistitem" may-throw-exception="no">
  + <function name="eelsbuildabsolutesize" may-throw-exception="no">
  + <function name="eelsbuildrelativeposition" may-throw-exception="no">
  + <function name="eelsbuildautosize" may-throw-exception="no">
  + <function name="eelsaddpanel" may-throw-exception="no">
  + <function name="eelsaddelement" may-throw-exception="no">
  + <function name="eelscreatedialog" may-throw-exception="no">
  + <function name="eelsaddpopupitementry" may-throw-exception="no">
  + <function name="eelscreateelementwithitems" may-throw-exception="no">
  + <function name="eelsbuildpositionfromapplication" may-throw-exception="no">
  </EELS_Functions>
```

http://hrem.mpi-stuttgart.mpg.de/koch/Vorlesung/Script/DM_PluginFunctions.html

Measuring Time



```
- <TimeFunctions>
  + <function name="GetHighResTickResolution" may-throw-exception="no" thread-safe="yes">
  + <function name="CalcHighResSecondsBetween" may-throw-exception="no" thread-safe="yes">
  + <function name="GetHighResTicksPerSecond" may-throw-exception="no" thread-safe="yes">
  + <function name="GetHighResTickCount" may-throw-exception="no" thread-safe="yes">
  + <function name="GetOSTickResolution" may-throw-exception="no" thread-safe="yes">
  + <function name="CalcOSSecondsBetween" may-throw-exception="no" thread-safe="yes">
  + <function name="GetOSTicksPerSecond" may-throw-exception="no" thread-safe="yes">
  + <function name="GetOSTickCount" may-throw-exception="no" thread-safe="yes">
  + <function name="GetCurrentTimeAndHighResTickCount" may-throw-exception="no" thread-safe="yes">
  + <function name="DeconstructUTCDate" may-throw-exception="no" thread-safe="yes">
  + <function name="ConstructUTCDate" may-throw-exception="no" thread-safe="yes">
  + <function name="DeconstructLocalGregorianDate" may-throw-exception="no" thread-safe="yes">
  + <function name="ConstructLocalGregorianDate" may-throw-exception="no" thread-safe="yes">
  + <function name="ParseTimeString" may-throw-exception="no" thread-safe="yes">
  + <function name="FormatTimeString" may-throw-exception="no" thread-safe="yes">
  + <function name="AddTimeUnitsToTime" may-throw-exception="no" thread-safe="yes">
  + <function name="CalcTimeUnitsBetween" may-throw-exception="no" thread-safe="yes">
  + <function name="GetUnixEpochTime" may-throw-exception="no" thread-safe="yes">
  + <function name="GetCurrentTime" may-throw-exception="no" thread-safe="yes">
  </TimeFunctions>
```