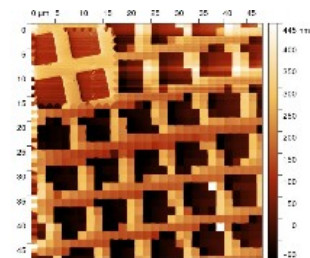
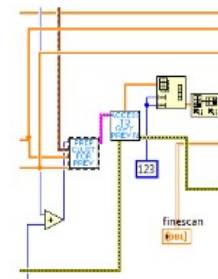
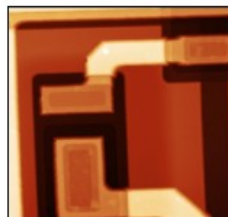
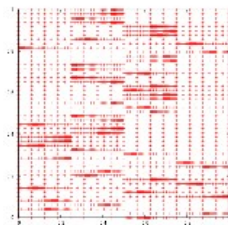
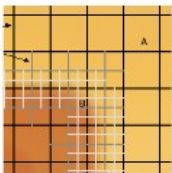
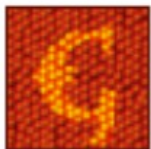


Gwyscan – library for smart scanning paths in SPM

Petr Klapetek, Andrew Yacoot, Petr Grolich, Miroslav Valtr, David Nečas

Czech Metrology Institute, Brno, Czech Republic
National Physical Laboratory, Teddington, Middlesex, UK
CEITEC, Masaryk University, Brno, Czech Republic



Speed is still big problem in most of the SPM systems, either custom built or commercial:

See the advertisements:

World's fastest AFM

Highest resolution fast scanning AFM

Ultra speed AFM

Speed is still big problem in most of the SPM systems, either custom built or commercial:

See the advertisements:

World's fastest AFM

Highest resolution fast scanning AFM

Ultra speed AFM

Speed is what sells AFM, or at least manufacturers think so.

There are two areas where this issue is really critical:

Speed is still big problem in most of the SPM systems, either custom built or commercial:

See the advertisements:

World's fastest AFM

Highest resolution fast scanning AFM

Ultra speed AFM

Speed is what sells AFM, or at least manufacturers think so.

There are two areas where this issue is really critical:

Large area measurements: measurements over millimeter areas with nanometer resolution, amount of data collected and also maximum speed of probe are limits.

Metrology SPMs: measurements with a lot of sensors, stopping in every point, Tradeoff between averaging and drift solved by better materials and even more sensors.

Reduction of time necessary for scanning?

Reduction of amount of data to save?

→ adaptive scanning

By adaptive scanning we can collect only what we really need.

Conventional AFM data

$z_{00} \ z_{01} \ z_{02} \ z_{03} \ \dots$

$z_{10} \ z_{11} \ z_{12} \ z_{13} \ \dots$

$z_{20} \ z_{21} \ z_{22} \ z_{23} \ \dots$

\dots

General XYZ data

$x_0 \ y_0 \ z_{0a} \ z_{0b} \ \dots \ t_0$

$x_1 \ y_1 \ z_{1a} \ z_{1b} \ \dots \ t_1$

$x_2 \ y_2 \ z_{2a} \ z_{2b} \ \dots \ t_2$

\dots

Conventional approach is however much simpler to handle and implement:

- fast visualisation
- many tools to process and evaluate the data

What do we need?

A software library with everything you need to build SPM with more complex scanning paths and general XYZ data as primary type of stored information.

Creating scan path for SPM, with different levels of complexity, user selections or automated path estimation.

Rendering **preview in conventional raster regime** to enable interaction with user during measurements.

Easy **saving data in Gwyddion compatible formats** - .gwy for any complex data including raster preview and .gxyzf for the simple XYZ data.

Support in Gwyddion for loading and managing the XYZ data and further processing them.

All this being **simple and compatible** with as many instruments as possible.

Primary implementation is in C
as a dynamically linked library it can
be interfaced from any language or
environment.

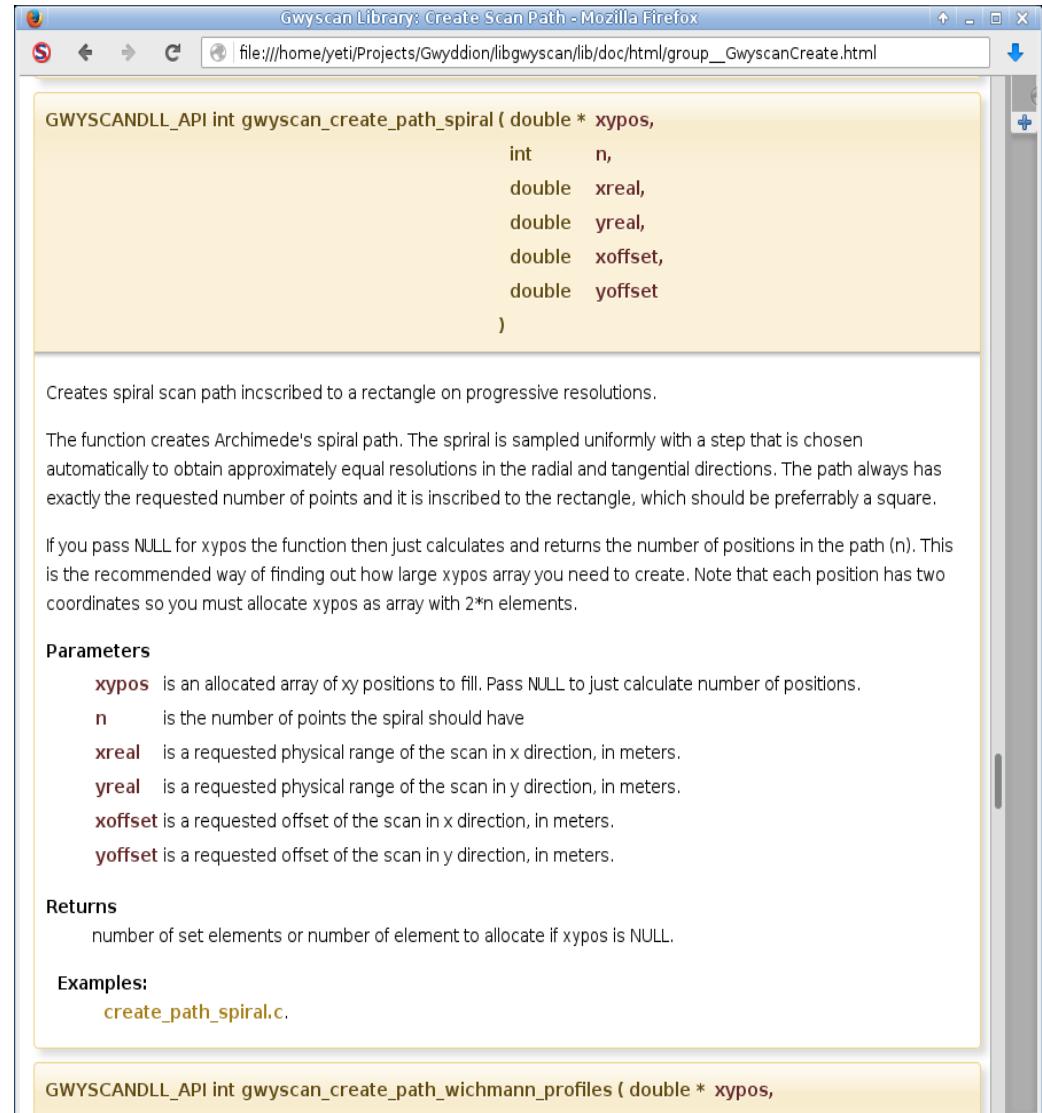
Basic function types:
create_scan_path_...

refine_path_...

preview_...

save_gwyddion_...

Saving data is based on much more
general **Gwyfile library** developed by
David Nečas:
<http://libgwyfile.sourceforge.net>



Gwyddion Library: Create Scan Path - Mozilla Firefox

file:///home/yeti/Projects/Gwyddion/libgwyddion/lib/doc/html/group__GwyddionCreate.html

```
GWYSCANDLL_API int gwyscan_create_path_spiral ( double * xypos,
                                                int      n,
                                                double  xreal,
                                                double  yreal,
                                                double  xoffset,
                                                double  yoffset
                                                )
```

Creates spiral scan path inscribed to a rectangle on progressive resolutions.

The function creates Archimede's spiral path. The spiral is sampled uniformly with a step that is chosen automatically to obtain approximately equal resolutions in the radial and tangential directions. The path always has exactly the requested number of points and it is inscribed to the rectangle, which should be preferably a square.

If you pass NULL for xypos the function then just calculates and returns the number of positions in the path (n). This is the recommended way of finding out how large xypos array you need to create. Note that each position has two coordinates so you must allocate xypos as array with 2*n elements.

Parameters

- xypos** is an allocated array of xy positions to fill. Pass NULL to just calculate number of positions.
- n** is the number of points the spiral should have
- xreal** is a requested physical range of the scan in x direction, in meters.
- yreal** is a requested physical range of the scan in y direction, in meters.
- xoffset** is a requested offset of the scan in x direction, in meters.
- yoffset** is a requested offset of the scan in y direction, in meters.

Returns

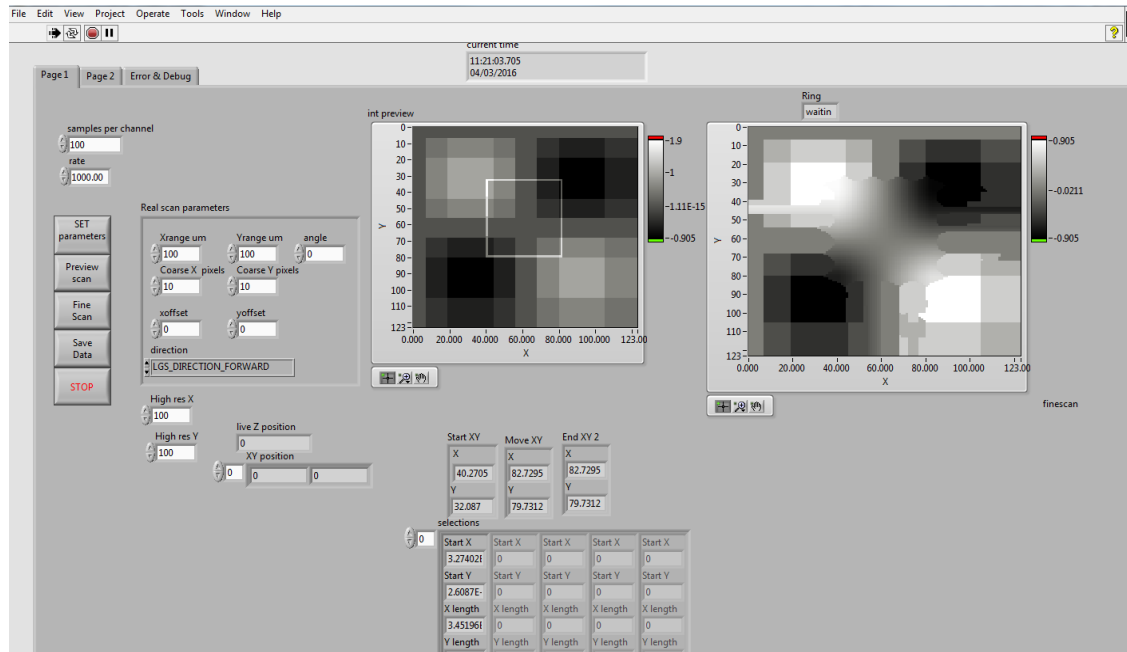
number of set elements or number of element to allocate if xypos is NULL.

Examples:

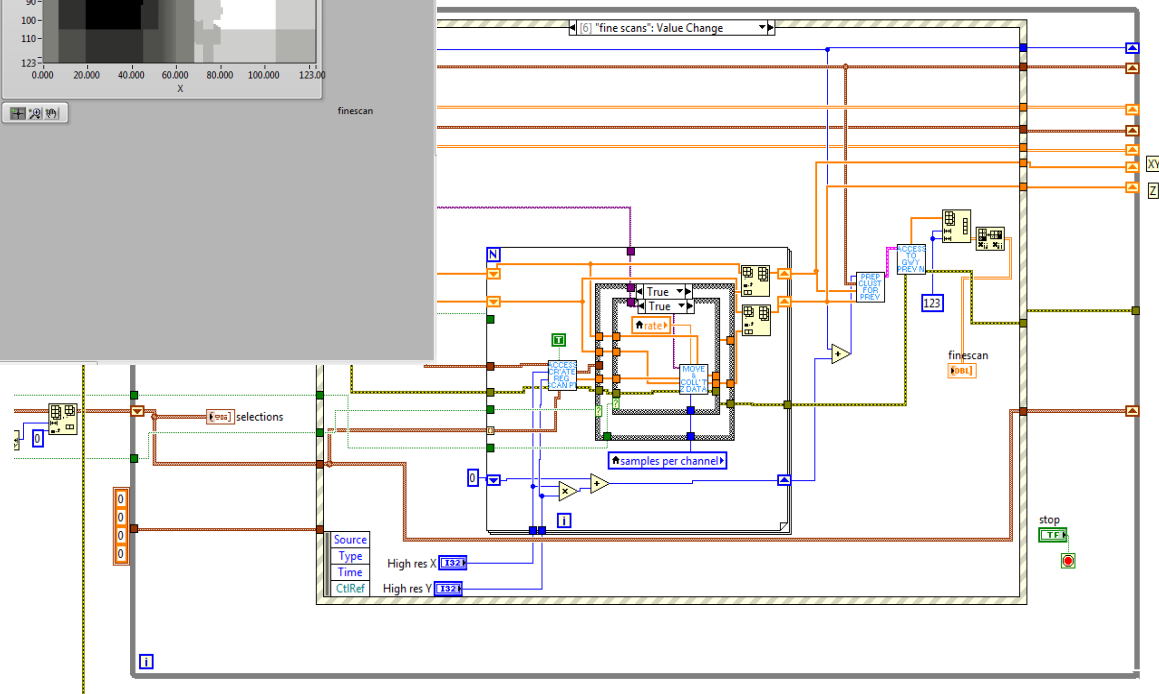
```
create_path_spiral.c.
```

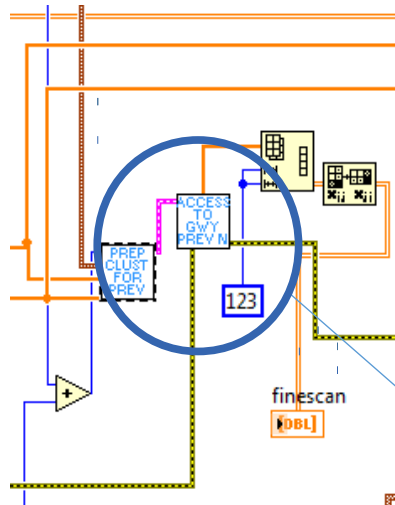
```
GWYSCANDLL_API int gwyscan_create_path_wichmann_profiles ( double * xypos,
```

Envelope for using the library within **Labview environment**, as a dynamically linked library.
Example: front panel and code for scanning



LabVIEW VI for
Gwyscan routine

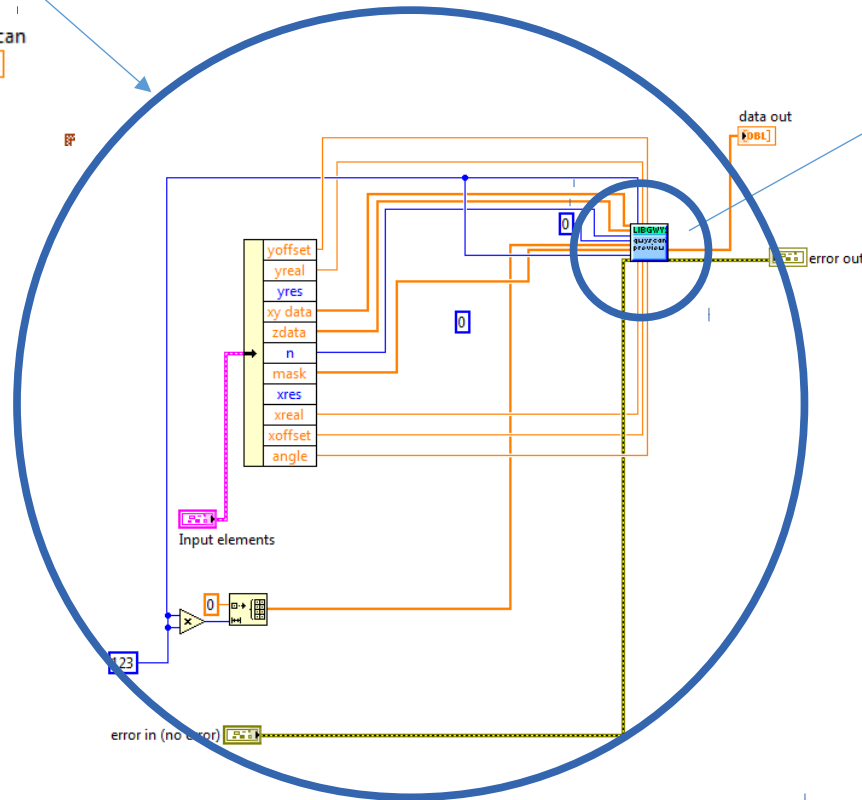




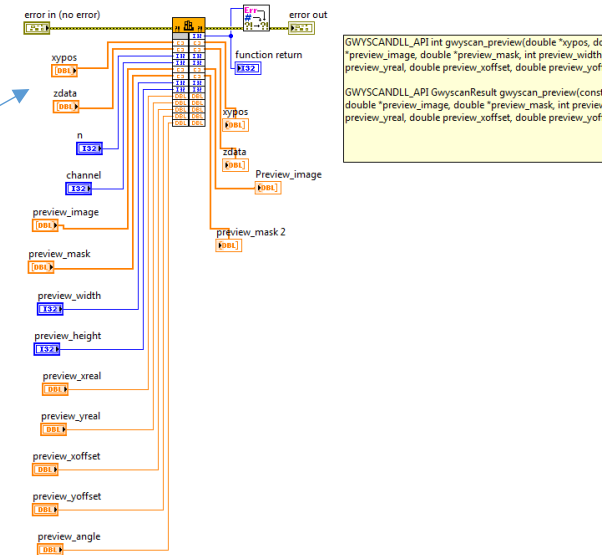
LabVIEW VI

Simplification of access via LabVIEW

LabVIEW code to sort out transfer of parameters via one cluster of parameter



Gwyscan routine and unpacking of cluster



GWYSCANDLL_API int gwyscan_preview(double *xypos, double *preview_image, double *preview_mask, int preview_width, double preview_xreal, double preview_yreal, double preview_xoffset, double preview_yoffset, double preview_angle)

GWYSCANDLL_API GwyscanResult gwyscan_preview(const double *xypos, double *preview_image, double *preview_mask, int preview_width, double preview_xreal, double preview_yreal, double preview_xoffset, double preview_yoffset, double preview_angle)

Simple AFM built from scratch for testing the routines on a independent system:

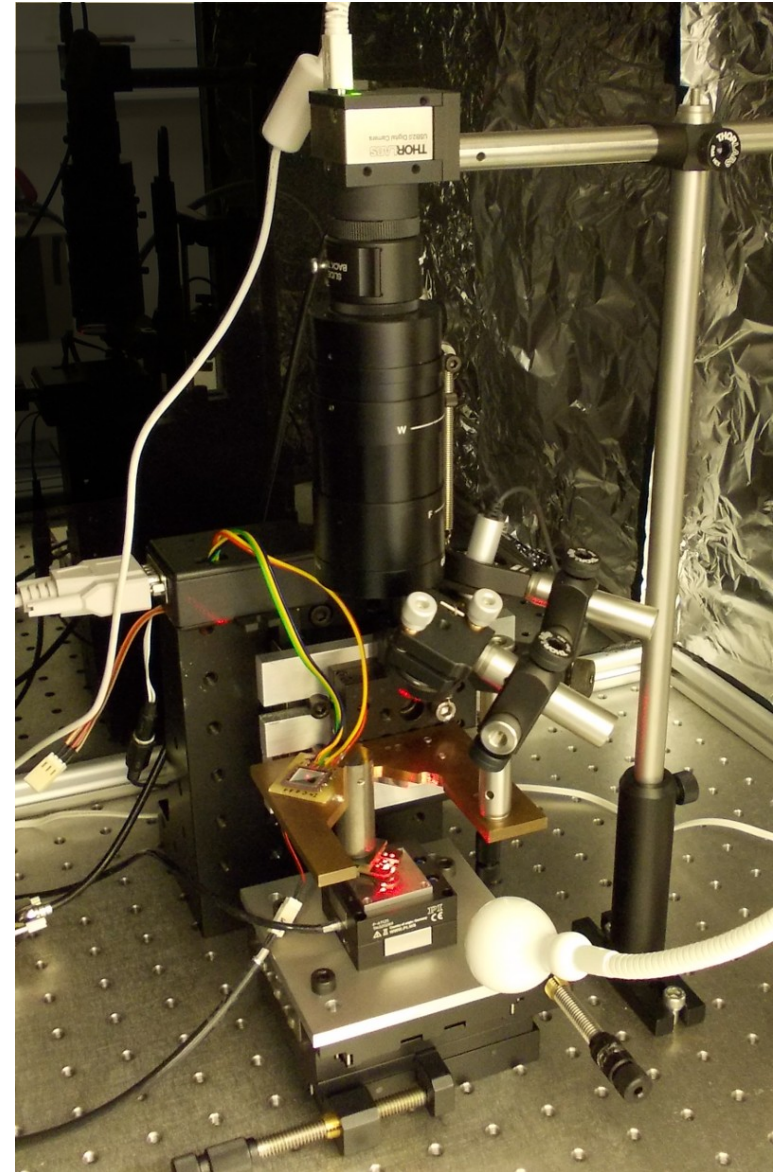
- PI stage (100x100 μm)
- custom built head
- analog feedback

Splitting of hardware part and user interface.

AFM is controlled by simple command line software acting as **server**. It can do only few commands:

- move table
- approach/retract
- report actual readings

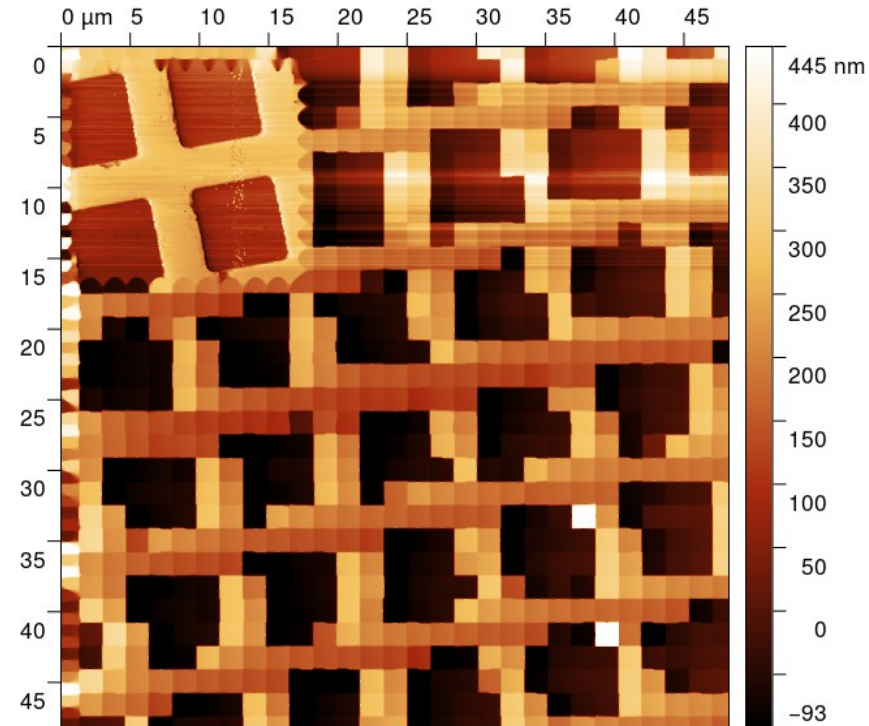
Client, connected via TCP socket, controls the scanning process.



Client can be written in any programming language supporting network communication.

Benefits of the system:

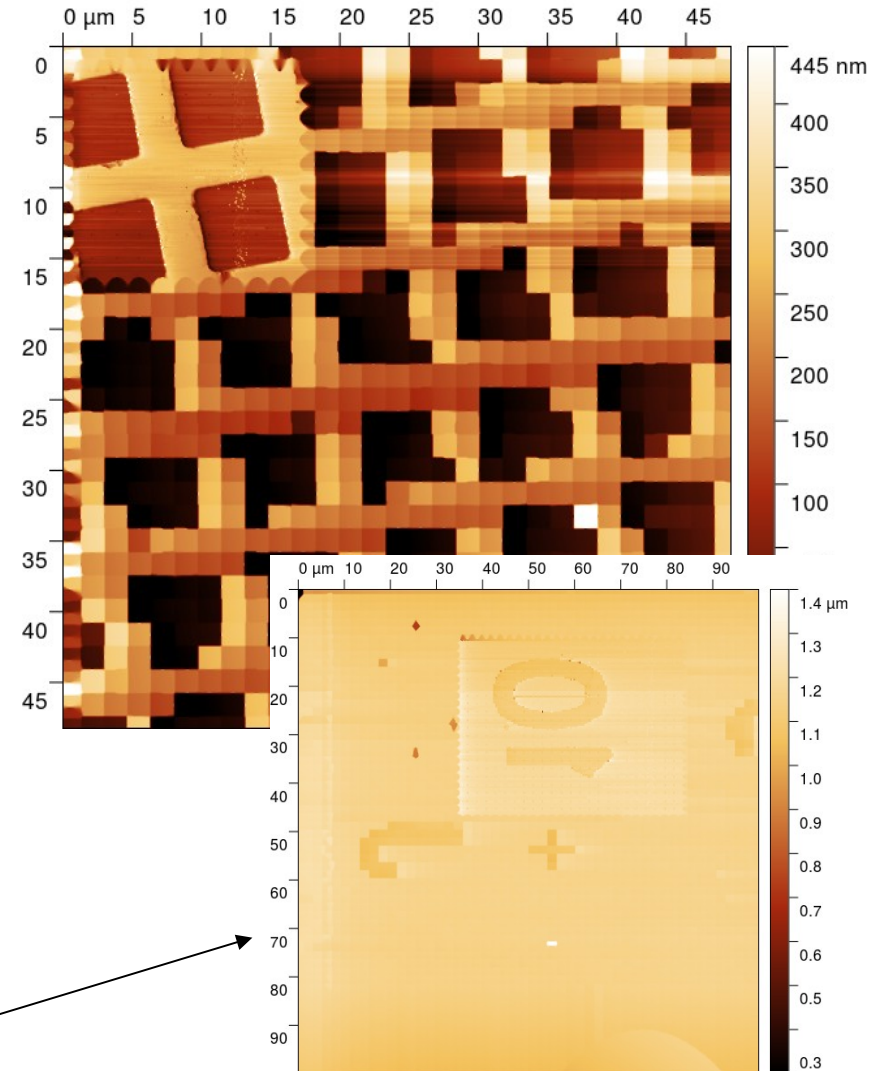
- simple design and operation
- independent on any previous HW and SW solution (e.g. NPL or CMI metrology SPM)
- observable drift even on samples with pronounced topography: good for benchmarking



Client can be written in any programming language supporting network communication.

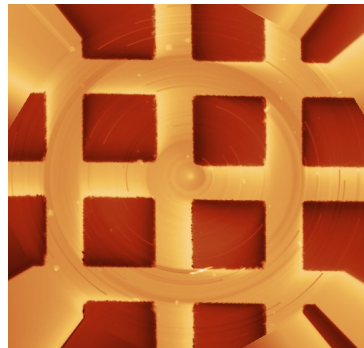
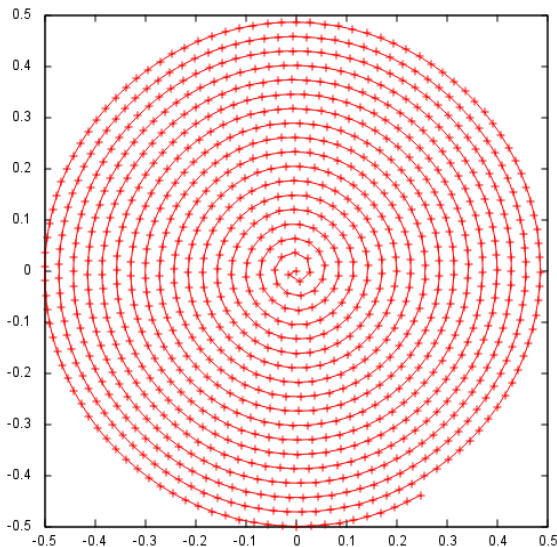
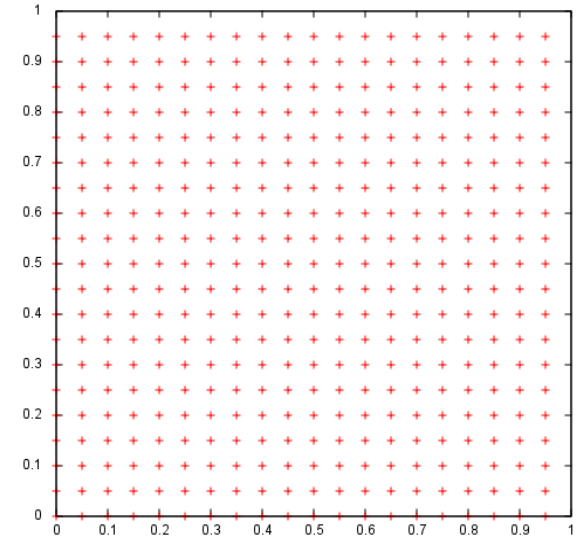
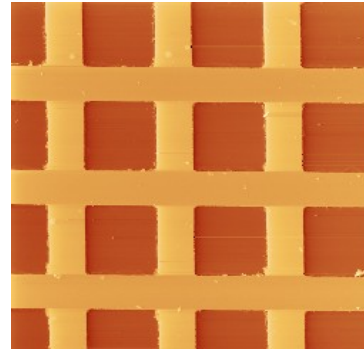
Benefits of the system:

- simple design and operation
- independent on any previous HW and SW solution (e.g. NPL or CMI metrology SPM)
- observable drift even on samples with pronounced topography: good for benchmarking



Regular scan path

- conventional approach
- one or bidirectional
- top-down or successively refined

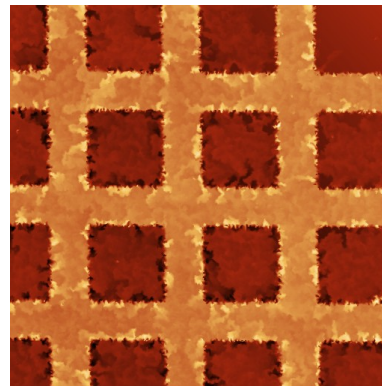
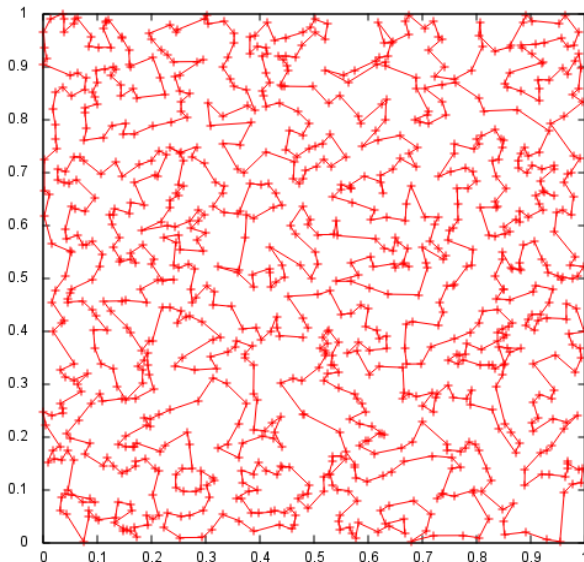
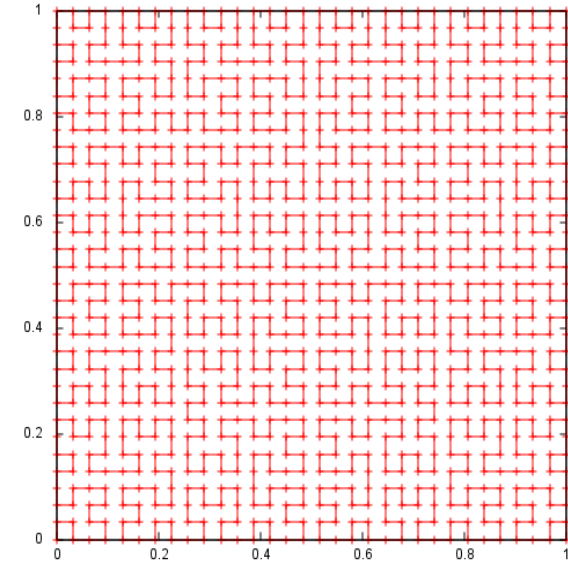
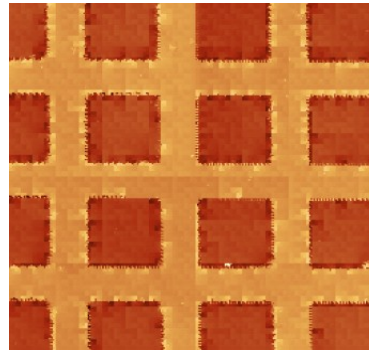


Spiral scan path

- used in some high-speed systems
- harmonic piezo movement

Space filling curve scan path:

- no fast and slow scanning axis
- can be successively refined

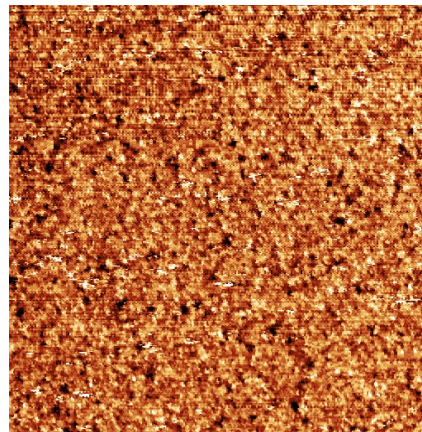
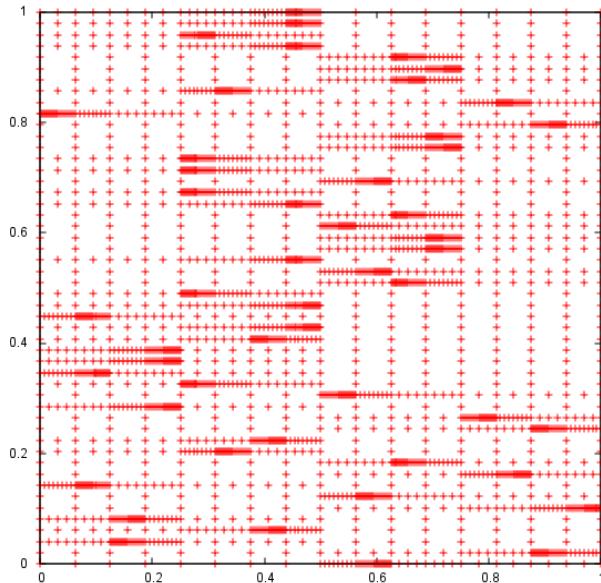
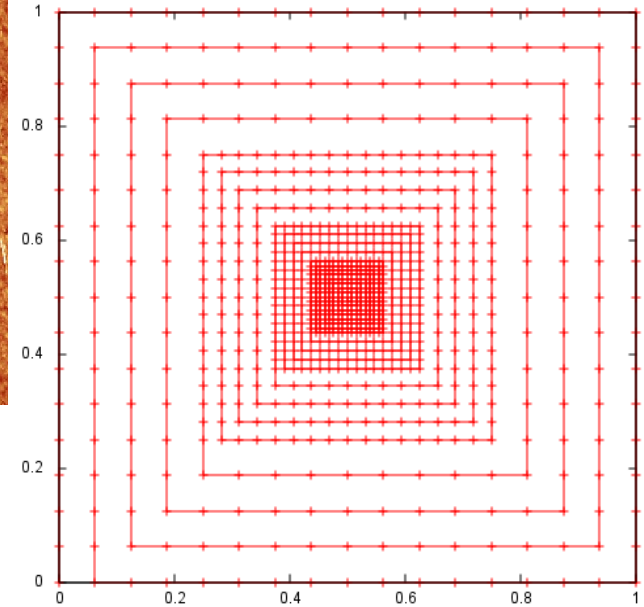
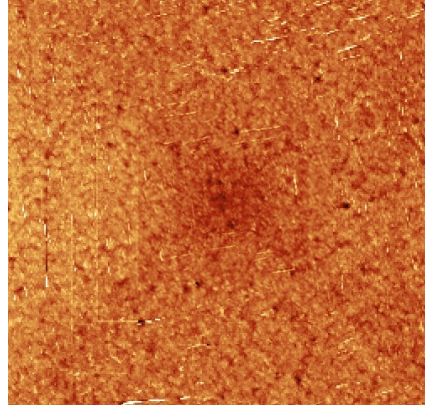


Random scan path

- randomly placed points
- traveling salesman problem

2D octave scan path

- for 2D statistical analysis
- multiple levels of refinement
- reduced fast/slow scan axis effect
- to be used e.g. with 2D FFT

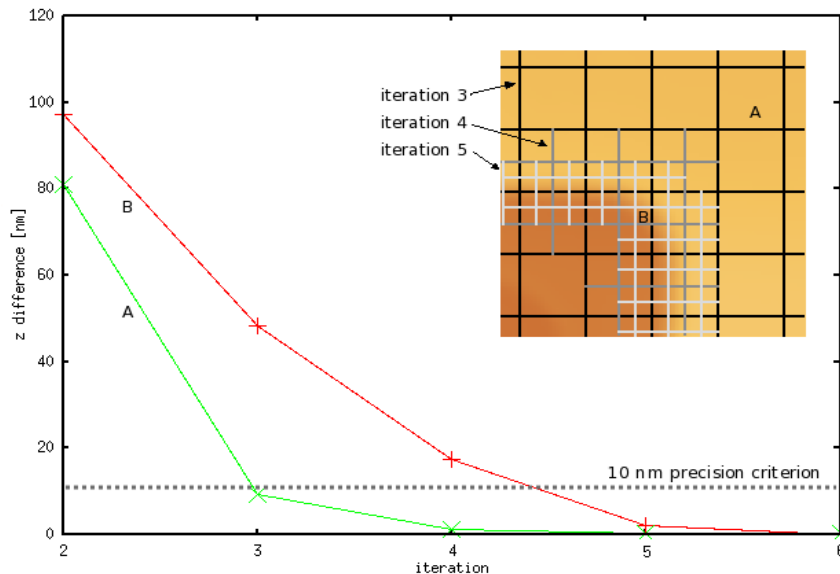
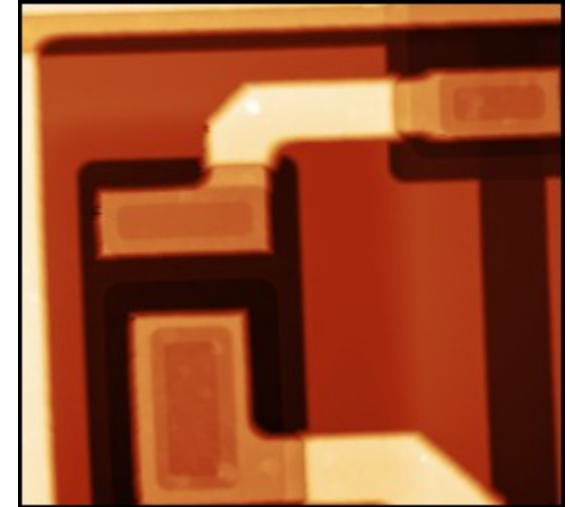


1D octave scan path

- for 1D statistical analysis
- better coverage of refined areas with randomised positions
- to be used e.g. with 1D FFT

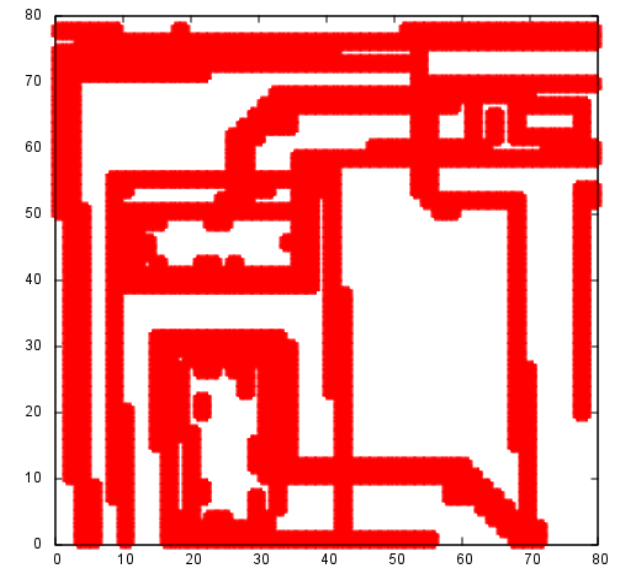
Threshold path refinement

- two previous measurements at coarse and finer level
- search for areas where the **refinement led to new information**
- further refinement in these areas



Local variation path refinement

- one previous measurement at coarse level
- search for areas where **local value of surface roughness (S_q) is bigger than a threshold value**
- further refinement in these areas



User selection path refinement

- user draws a selection on a preview to be refined

Drift becomes more visible on most of the data:

However, it exists also on regular data.

More visible = easier to distinguish.

Drift becomes more visible on most of the data:

However, it exists also on regular data.

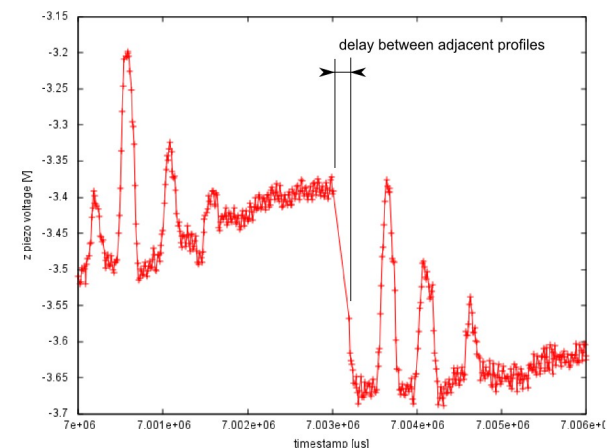
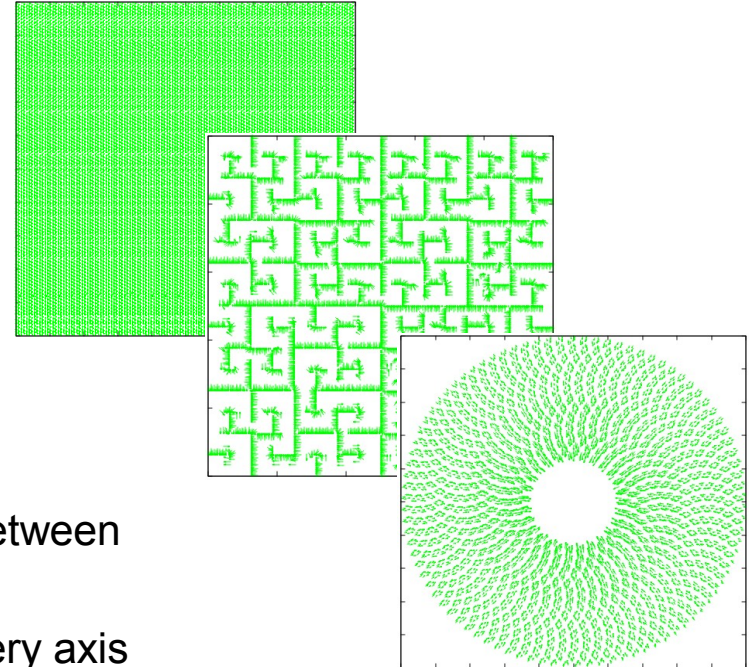
More visible = easier to distinguish.

Procedure for drift estimation and correction:

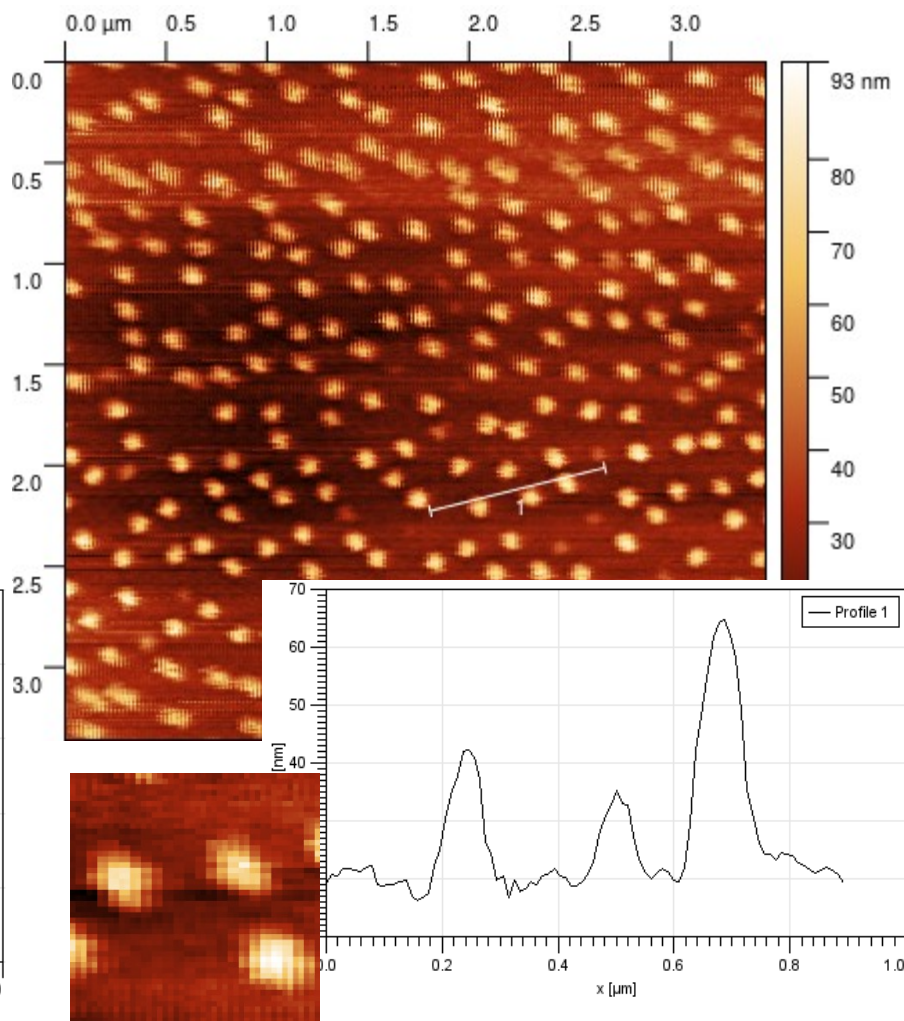
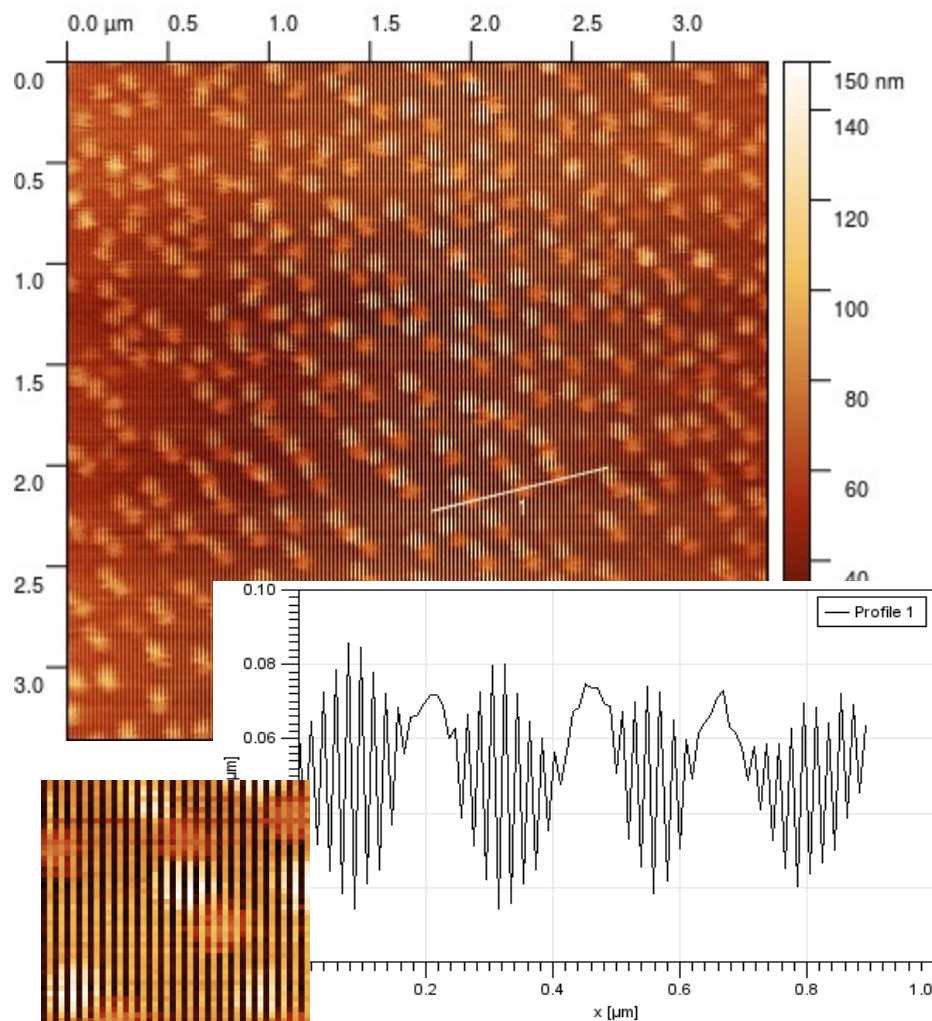
- find all the neighbor or identical pixels measured in significantly different time
- estimate the x, y, and z drift to minimize differences between z values at these pixels:
 - create model of the drift dependence on time for every axis
 - evaluate the sum of squared differences
 - modify the model of the drift

The same algorithm can be used both for single and repeated measurements, and is independent on scanning path.

Key idea: use timestamp for fitting the drift dependences



Drift correction in practice: linear drift compensation in all the axes



Gwyscan is a cross-platform library for handling arbitrary scanning paths in SPM. It was tested to work with C and Labview programming environments, and at least for these two environments the documentation is being prepared.

It covers planning, visualisation and data storage related with many different types of scanning paths.

Main data type are XYZ data: x, y, z_a, z_b, \dots , (+timestamp).

It allows easy refinement of the scan at desired locations.

There are also special scanning paths suitable for particular purposes, e.g. local variation analysis, these scanning paths have statistical properties that are good for evaluation algorithms.

Complex scanning paths can be also good for estimating drift in SPM from single measurement.

Read more: Meas. Sci. Technol. 28 (2017) 034015

Try it: <http://libgwyscan.sf.net>