BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF MECHANICAL ENGINEERING

Fakulta strojního inženýrství

INSTITUTE OF PHYSICAL ENGINEERING

ÚSTAV FYZIKÁLNÍHO INŽENÝRSTVÍ

TEMPERATURE DRIFT COMPENSATION FOR NANOSTRUCTURE ANALYSIS

KOMPENZACE TEPLOTNÍHO DRIFTU PŘI ANALÝZE NANOSTRUKTUR

BACHELOR'S THESIS

BAKALÁŘSKÁ PRÁCE

AUTHOR AUTOR PRÁCE	STANLI HAKIRA
ADVISOR VEDOUCÍ PRÁCE	Ing. PETR BÁBOR, PhD.

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ

Zadání bakalářské práce

Ústav:	Ústav fyzikálního inženýrství
Student:	**Stanli Hakira**
Studijní program:	Aplikované vědy v inženýrství
Studijní obor:	Fyzikální inženýrství a nanotechnologie
Vedoucí práce:	**Ing. Petr Bábor, Ph.D.**
Akademický rok:	2017/18

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Kompenzace teplotního driftu při analýze nanostruktur

Stručná charakteristika problematiky úkolu:

V současné době firma Tescan ve spolupráci s Ústavem fyzikálního inženýrství vyvíjí ultravakuový rastrovací elektronový mikroskop (UHV SEM). Byla navržena UHV komora, do které byl úspěšně instalován elektronový tubus, který byl adaptován pro UHV. Toto zařízení musí umožňovat tvorbu a charakterizaci nanostruktur. Při tvorbě nanostruktur je často nutné ohřívat vzorek na vysoké teploty a současně provádět depozici materiálu na vzorek. Zařízení UHV SEM má umožnit nejenom tvorbu nanostruktur, ale i in–situ sledování této tvorby pomocí elektronového mikroskopu. Pří sledování vznikající nanostruktury dochází k jejímu pohybu v důsledku tepelné roztažnosti vzorku a jeho držáku (tzv. teplotní drift). V důsledku toho je obtížné dané struktury udržet v zorném poli mikroskopu a je nutná kompenzace teplotního driftu. Z těchto důvodů je třeba v bakalářské práci vyvinout metody, které by vliv driftu maximálně potlačily.

Cíle bakalářské práce:

1. Proveďte rešerši možností kompenzace teplotního driftu při pozorování v SEM.
2. Navrhněte a realizujte metodu, která by teplotní drift dokázala kompenzovat.
3. Proveďte experiment, který prokáže funkčnost této metody.

Seznam doporučené literatury:

CUI, Z., Nanofabrication, Principles, Capabilities and Limits, Springer, 2008.

REIMER, L., Scanning Electron Microscopy, Physics of Image Formation and Microanalysis, Springer, 1998.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2017/18

V Brně, dne 9. 11. 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ
DĚKANÁT
11

L. S.

prof. RNDr. Tomáš Šikola, CSc.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstract

An ultra high vacuum apparatus for nanostructure experiments is being developed by the Tescan company in cooperation with the Institute of Physical Engineering of the Faculty of Mechanical Engineering. The apparatus is designed for preparation, modification, and analysis of nano-scale structures. A sample holder which allows heating and cooling has been developed for the apparatus. A scanning electron microscope is attached to the chamber to provide analytical and manufacturing capabilities. During experiments with heating enabled, the sample moves relative to the SEM column, causing drift of the image. This bachelor thesis proposes a solution to the problem of temperature drift by the means of motion tracking based on image registration using Fourier transform. An application complementary to the SEM control software which implements the algorithm was designed and tested at the instrument.

Keywords

drift compensation, scanning electron microscope, SEM, image registration, phase correlation, nanostructure

Abstrakt

Ve spolupráci mezi firmou Tescan a Ústavem fyzikálního inženýrství fakulty Strojního inženýrství je vyvíjena ultravakuová aparatura pro experimenty v nanotechnologii. Komora je navržena na přípravu, modifikaci a analýzu nanostruktur. Pro komoru byl vyroben držák na vzorky umožňující zahřívání a chlazení vzorku. Na komoře je instalován skenovací elektronový mikroskop sloužící k analytickým a výrobním technikám. V průběhu zahřívaní při dochází k posunu obrazu vzorku vzhledem k tubusu SEMu, což zprůsobuje drift obrazu. Tato bakalářská práce navrhuje kompenzaci teplotního driftu pomocí algoritmu sledujicího polohu pužitím obrazové registrace založené na Fourierově transformaci. Byla navržena a implementována aplikace doplňující ovládací software SEMu, která byla prakticky otestována.

Klíčová slova

kompenzace posunu, skenovací elektronový mikroskop, SEM, obrazová registrace, fázová korelace, nanostruktura

HAKIRA, Stanli. *Temperature drift compensation for nanostructure analysis.* Brno: Brno University of Technology, Faculty of Mechanical Engineering, 2018. 21 pages. Advisor Ing. Petr Bábor, Ph.D.

I declare that I have written the bachelor's thesis ”Temperature drift compensation for nanostructure analysis” on my own according to advice of my bachelor's thesis advisor Ing. Petr Bábor, Ph.D., and using the sources listed in references.

May 4, 2018

Stanli Hakira

Acknowledgments

I would like to thank Ing. Petr Bábor, Ph.D. for his advice on this bachelor's thesis, and for professional and friendly approach. I would also like to thank Bc. Tomáš Axman for helping me get familiar with the apparatus and prepare the experiment.

I would also like to express my thanks to prof. RNDr. Miloslav Druckmüller, CSc. for pointing me in the direction of image registration based on phase correlation and literature recommendation.

Sincere thanks to Tescan Brno, s.r.o. and their staff for providing an amicable environment.

Contents

1 Introduction

Many modern technologies depend on the experience obtained by researching nanotechnology. In general, nanostructures require very specific instruments to be manufactured and observed. To enable researchers to work effectively, high quality instruments fulfilling the requirements for reliability, usability, and nano-scale parameters have to be used.

During a visit at the Tescan company in Brno, I was intrigued by the project developing an ultra-high-vacuum (UHV) experimental apparatus with a scanning electron microscope (SEM) in cooperation with the Institute of Physical Engineering of the Faculty of Mechanical Engineering. Therefore, I gladly accepted the opportunity to focus on a practically applicable topic during my bachelor's thesis and take part in improving the instrument.

A sample holder for measuring electrical properties in a wide range of temperatures was developed for the apparatus, which can be used to heat the sample up to 700 K or cool it down to 20 K, the development of which is described in [2]. When heated, the sample exhibits temperature drift, that is, it moves relative to the view field of the SEM, which is likely caused by thermal expansion and contraction of the palette as well as the sample.

In the thesis, I propose a programmatic solution for temperature drift compensation and design an application complementary to the standard control software used for the UHV SEM, which helps to improve the usability during measurements within a wide range of temperatures.

The standard control software does not implement mechanisms for drift correction, which was causing any temperature drift to proceed unchecked, resulting in the object drifting out of the view field, devaluing the measurement. The researcher would be inevitably forced to remain constantly monitoring the image and manually intervene to correct for any potential drift.

Another advantage of the application is the ability to record video of the experiment in conjunction with position information, which can later help with interpreting the experiment.

Approaching the task at hand, I was trying to find the most effective and uncomplicated way of compensating temperature drift. Considering the structures may vary widely (size, shape and background contrast are not clearly defined) and develop in time, the use of tracking algorithms based on feature detection might not always yield satisfactory results. I decided to use shift detection based on image registration. This method is commonly used to stitch multiple images into a larger continuous image.

Initial chapters are dedicated to description of the apparatus for which the application is designed, and the theory behind image registration. In the following chapters I propose an algorithm for drift compensation and implement an application, which communicates with the SEM, compensates drift, and records the experiment. Final chapter practically verifies the functionality of the proposed algorithm and the application at the UHV SEM.

2 Description of the apparatus

The apparatus is developed in cooperation among Tescan, Institute of Physical Engineering of the Faculty of Mechanical Engineering, and the Institute of Scientific Instruments of the Czech Academy of Sciences as a part of the Amispec project. The resulting apparatus is planned to provide various methods for nanostructure manufacturing and analysis. Due to requirements for high cleanliness in the apparatus, all chambers use ultra-high vacuum.

During the thesis, I worked on a prototype situated in a laboratory at Tescan. Mounting frame for the apparatus stands on adjustable rubber feet. The main experimental chamber rests on piezoelectric dampeners atop of the frame, which provides active vibration reduction. Figure 1 depicts the apparatus as configured at the time of writing. Various instruments are connected to the chamber, the most important of which for my work is the scanning electron microscope column mounted on top of the chamber and a secondary electron detector mounted on the side. Others include a loading chamber separated by a desk valve, a deposition chamber with effusion cells, a vacuum pump, a Bayard-Alpert vacuum gauge, and a digital camera providing a view of the main chamber. A rack containing electronics for the apparatus is situated next to the frame.

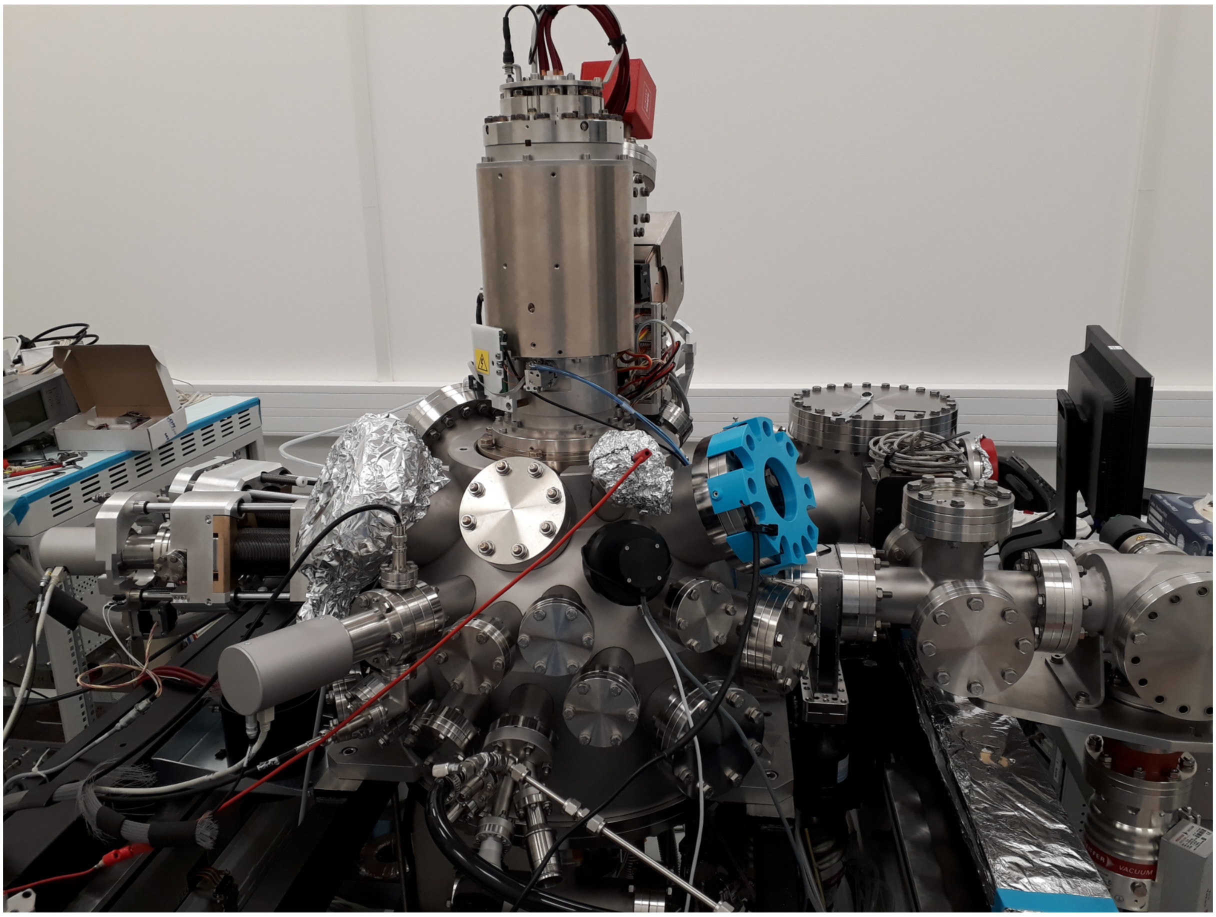


Figure 1: Overview of the experimental apparatus

2.1 SEM principle of operation

Electrons are emitted from a cathode[1] and accelerated using a high voltage usually ranging from 100 V to 50 kV to create a beam. The beam is reduced to proper cross section using an aperture and condenser lenses. The condensed beam travels through objective lens which focuses the beam onto the sample. Scanning coils deflect the beam to a particular point on the sample. [13]

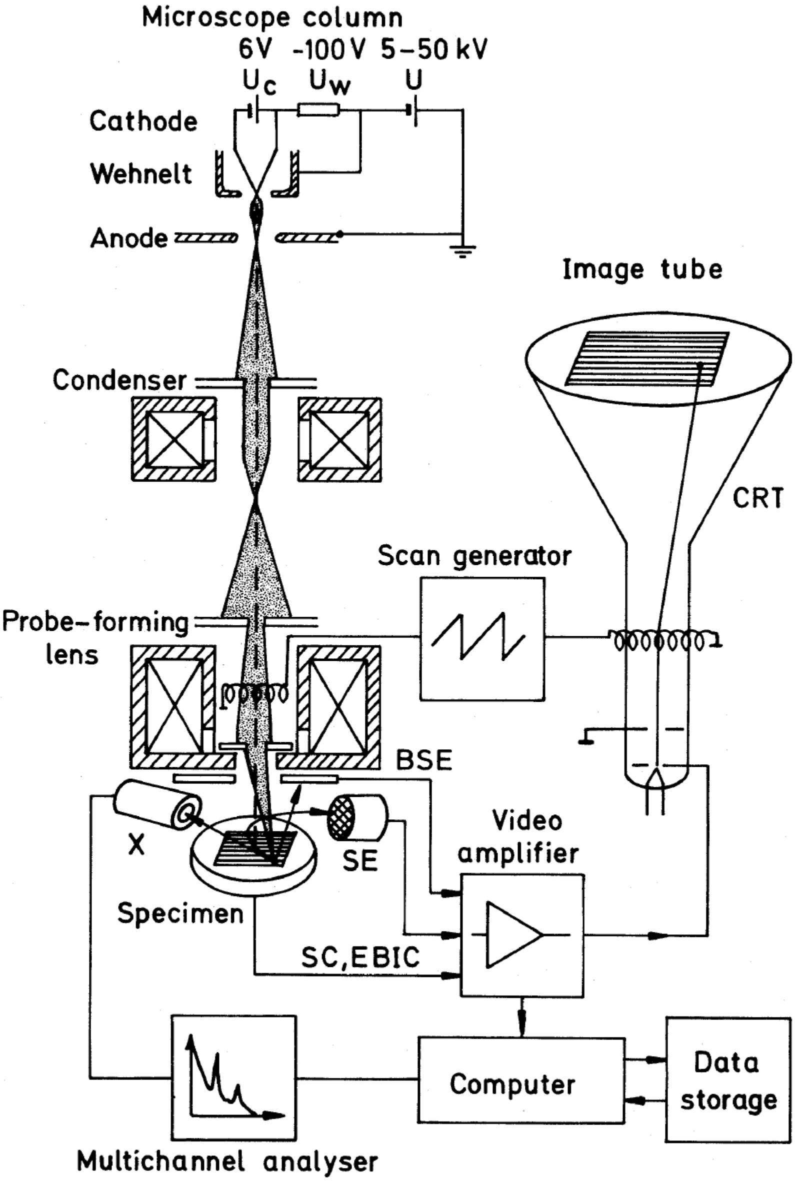


Figure 2: Schematic of a scanning electron microscope, from [13, page 2]

As the high energy primary electron beam hits the sample, various effects can be observed depending on the energy of the primary electrons. Figure 3 shows a comparison of different types of electrons which can be detected. Secondary electrons are the most important for imaging topography; they provide the highest resolution, because they are emitted from several surface layers when a primary electron colides with an electron in the shell of an atom giving it enough energy to leave the shell and become a secondary electron.

[1] Cathodes can be based on thermal emission, field emission, or Schottky effect.

Back-scattered electrons come from a region deeper within the sample and are caused by a primary electrons scattering on the atoms of the sample. Other effects such as high energy photons (characteristic and continuous x-rays) and Auger electrons can be observed as well.

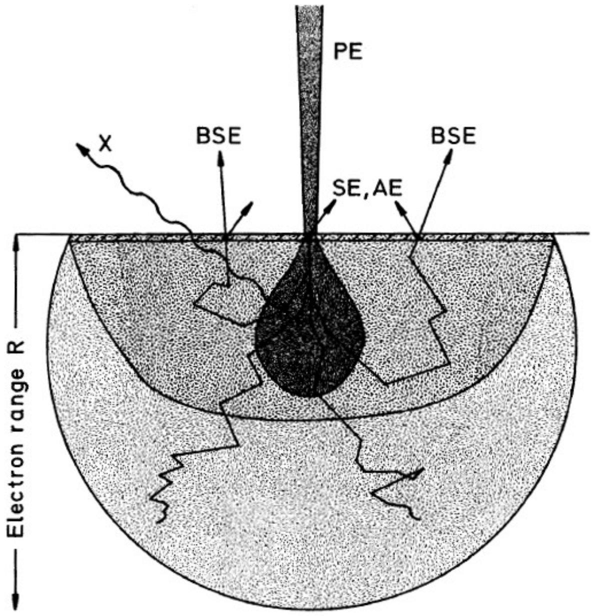


Figure 3: Interaction of electrons with a sample; PE – primary electrons, SE – secondary electrons, BSE – back-scattered electrons, x - high energy photons, from [13, page 5]

2.2 Imaging parameters

To properly control a scanning electron microscope, several imaging parameters must be understood. Scanning is done by deflecting the electron beam focused on a sample to follow a series of lines to fill a view field[2]. The view field is defined by the physical size (in millimeters), the raster size (in pixels), and the position relative to the center of the column, so called ”beam shift” (in millimeters). Beam shift is handled by scanning coils, making its positional accuracy comparable[3] to that of scanning.

Electron detectors commonly have only one pixel, thus it is necessary to image only one pixel at a time. While imaging, the scanning generator lets the beam remain at each particular point in the raster a given amount of time, referred to as ”dwell time” (in nanoseconds), as the detector counts signal pulses, which are transformed into the value of the corresponding pixel on the scanned image. This approach allows the detector to have a fixed solid angle view of the sample, resulting in a constant fraction of the electrons emitted from the sample to be detected. Higher dwell time thus results in a less noisy image (provided constant background noise), however, it also results in a slower scan speed.

[2] square view field is common, but in principle any shape composed of lines is possible

[3] strong aberrations are present outside of a particular range, some SEMs (like the Tescan model used) limit the beam shift range to prevent this

3 Image registration

Image registration provides means to compare two images in order to determine the way in which one image differs from the other. The algorithm based on phase correlation expects digital images that can differ only in shift[4]. The exposure, contrast, and the properties of noise have to be identical for both images.[3] This is an important consideration, as it implies a key requirement for the acquisition of images – the entirety of an image has to be scanned at a time without dramatic development of the structure in view. In addition, both images have to be taken by the same sensor and there must not be a significant development between the images. A limitation is also that the maximum shift which can possibly be detected is $\left(\frac{N}{2}, \frac{M}{2}\right)$. Therefore, as long as the drift caused by changing temperature occurs significantly faster than processes on the structure and slower than scanning speed, this approach should provide a sufficient way to determine the shift and compensate drift between the images.

3.1 Image

Detectors from the SEM provide gray-scale digital images which are usually 8-bit or 16-bit. Such an image can be mathematically represented using a two-dimensional matrix (a discrete function) $\boldsymbol{W} : \{0, 1, \ldots, N-1\} \times \{0, 1, \ldots, M-1\} \to \{0, 1, \ldots, 2^n-1\}, N, M \in \mathbb{N}$. Sometimes images are stored as real matrices during computation $\boldsymbol{W}(x, y) \in \mathbb{R}$. The number of different values in a pixel n is determined by the number of bits of the image. For 8-bit images $2^8 = 256$ and for 16-bit images $2^{16} = 6536$. Matrix elements outside of $x \in \{0, 1, \ldots, N-1\}$, $y \in \{0, 1, \ldots, M-1\}$ are undefined and can be assumed zero.

3.2 Principle of shift detection

Image registration based on phase correlation is used to detect the shift between images. As described in [3], phase correlation can be calculated using the Fourier transform. Let $\boldsymbol{A}(x, y), \boldsymbol{B}(x, y)$ be real images and let $\boldsymbol{\mathcal{A}}(\zeta, \eta)$, $\boldsymbol{\mathcal{B}}(\zeta, \eta)$ be their Fourier spectra. Let (x_0, y_0) be a shift vector between $\boldsymbol{A}$ and $\boldsymbol{B}$ such that $\boldsymbol{A}(x + x_0, y + y_0) = \boldsymbol{B}(x, y)$. The matrix $\boldsymbol{\mathcal{Z}}_{\boldsymbol{A},\boldsymbol{B}} : \{0, 1, \ldots, N-1\} \times \{0, 1, \ldots, M-1\} \to \mathbb{C}$ defined as

$$\boldsymbol{\mathcal{Z}}_{\boldsymbol{A},\boldsymbol{B}} = \frac{\boldsymbol{\mathcal{A}} \cdot \boldsymbol{\mathcal{B}}^*}{|\boldsymbol{\mathcal{A}} \cdot \boldsymbol{\mathcal{B}}|} = \frac{\boldsymbol{\mathcal{A}} \cdot \boldsymbol{\mathcal{A}}^* (\mathrm{e}^{-i(\zeta x_0 + \eta y_0)})^*}{|\boldsymbol{\mathcal{A}} \cdot \boldsymbol{\mathcal{A}} (\mathrm{e}^{-i(\zeta x_0 + \eta y_0)})|} = \mathrm{e}^{-i(\zeta x_0 + \eta y_0)}$$

is called the cross-power spectrum of matrices $\boldsymbol{A}$, $\boldsymbol{B}$.
The matrix $\boldsymbol{P}_{\boldsymbol{A},\boldsymbol{B}} : \{0, 1, \ldots, N-1\} \times \{0, 1, \ldots, M-1\} \to \mathbb{C}$ defined as

$$\boldsymbol{P}_{\boldsymbol{A},\boldsymbol{B}} = \mathcal{F}^{-1}\{\boldsymbol{\mathcal{Z}}_{\boldsymbol{A},\boldsymbol{B}}\}$$

is called the phase-correlation matrix[5], which provides information about the shift.

[4] It is also possible to detect rotation and scale-change using phase correlation, but it is not used here.

[5] The phase-correlation matrix is real, because both $\boldsymbol{A}$ and $\boldsymbol{B}$ are real. See theorem 3.27 in [3]. The absolute value of P may be used instead to guarantee this.

A global maximum in both x and y can be found at the coordinates $\boldsymbol{P}_{\boldsymbol{A},\boldsymbol{B}}(-x_0, -y_0)$. The shift vector can therefore be determined

$$(x_0, y_0) = -\arg\max\{\boldsymbol{P}\}$$

The matrix $\boldsymbol{P}^{p,q}_{\boldsymbol{A},\boldsymbol{B}} : \{0, 1, \ldots, N-1\} \times \{0, 1, \ldots, M-1\} \to \mathbb{C}$, $p, q \in \mathbb{R}_0^+$ defined as

$$\boldsymbol{P}^{p,q}_{\boldsymbol{A},\boldsymbol{B}} = \mathcal{F}^{-1}\left\{\frac{\boldsymbol{\mathcal{A}} \cdot \boldsymbol{\mathcal{B}}^*}{(|\boldsymbol{\mathcal{A}}| + p) \cdot (|\boldsymbol{\mathcal{B}}| + q)}\right\}$$

is called the semi-phase correlation matrix of matrices $\boldsymbol{A}$, $\boldsymbol{B}$ with parameters p, q. The semi-phase correlation can be used in place of the phase-correlation when some values of the image may be close to zero.

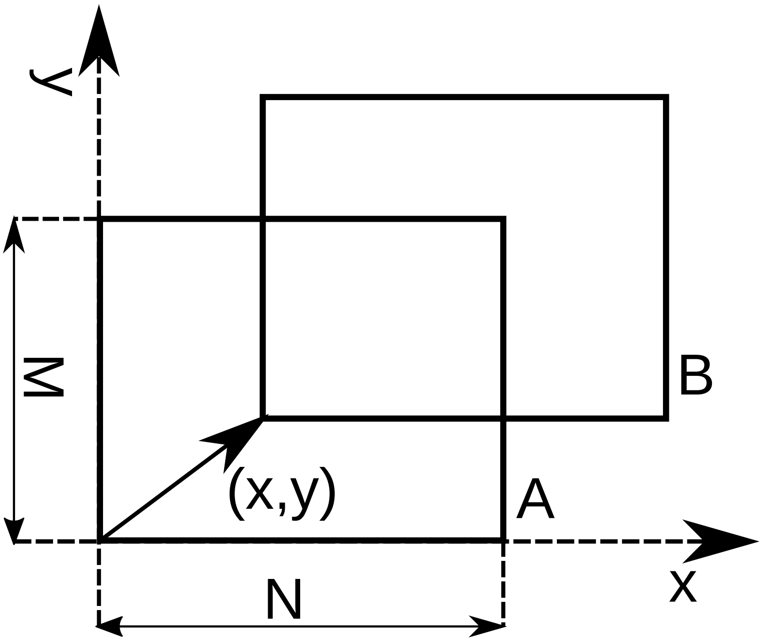


Figure 4: Illustration of shift between images

3.3 Sub-pixel resolution

By finding the maximum in the phase-correlation matrix, an integer shift vector can be determined. However, it is possible to calculate an estimate of the shift vector with sub-pixel resolution using more points around the maximum. A possible approach is to calculate image moments around the center as in [4]. The solution described below fits a quadratic function through the neighbors of (x_0, y_0), the same approach is discussed in [1],[3] and used by [6].

Let $\boldsymbol{F}$ be a sub-matrix of $\boldsymbol{P}$ such that

$$\boldsymbol{F} = \begin{pmatrix} \boldsymbol{P}(x_0-1, y_0-1) & \boldsymbol{P}(x_0, y_0-1) & \boldsymbol{P}(x_0+1, y_0-1) \\ \boldsymbol{P}(x_0-1, y_0) & \boldsymbol{P}(x_0, y_0) & \boldsymbol{P}(x_0+1, y_0) \\ \boldsymbol{P}(x_0-1, y_0+1) & \boldsymbol{P}(x_0, y_0+1) & \boldsymbol{P}(x_0+1, y_0+1) \end{pmatrix}$$

In case the maximum at (x_0, y_0) is on the edge of the phase-correlation matrix, the matrix is periodically expanded to determine the missing values.

$$\boldsymbol{P}'(x, y) = \boldsymbol{P}(|N + x| \bmod N, |M + y| \bmod M)$$

The first and second order derivatives of $\boldsymbol{F}$ are used to calculate the sub-pixel correction. Let $f_i = \frac{\partial \boldsymbol{F}(1,1)}{\partial i}$, $f_{ij} = \frac{\partial^2 \boldsymbol{F}(1,1)}{\partial i \partial j}$. The sub-pixel correction is then obtained as

$$(-\tilde{x}, -\tilde{y}) = \left(\frac{f_{yy} f_x - f_y f_{xy}}{(f_{xy})^2 - f_{xx} f_{yy}}, \frac{f_{xx} f_y - f_x f_{xy}}{(f_{xy})^2 - f_{xx} f_{yy}} \right).$$

The approximation of the shift vector with sub-pixel resolution is $(\bar{x}, \bar{y}) = (x_0 + \tilde{x}, y_0 + \tilde{y})$.

3.4 Filtering noise

In the case of correlating images from an electron microscope acquired at the limits of the device, it is likely that a substantial portion of the image will be obstructed by "additive noise" – random sensor noise that changes with every image. To minimize the effect of additive noise on the quality of image registration, a low-pass filter is used to reduce the noise.[3]

The spatial frequencies obtained from a Fourier spectrum of an image represent the rate of change of pixel values. A low-pass filter is a function reducing the magnitude of high frequencies in the frequency spectrum. A Gaussian low-pass filter will be used to ensure smooth transition in the spectrum.

Let $\sigma \in \mathbb{R}_0^+$. The function $G_\sigma(x, y) : \mathbb{R}^2 \to (0, 1]$ defined as

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \mathrm{e}^{-\frac{(x - x_c)^2 + (y - y_c)^2}{2\sigma^2}}$$

is referred to as the Gaussian low-pass weight function with standard deviation σ, where (x_c, y_c) is the center of the image.[15]

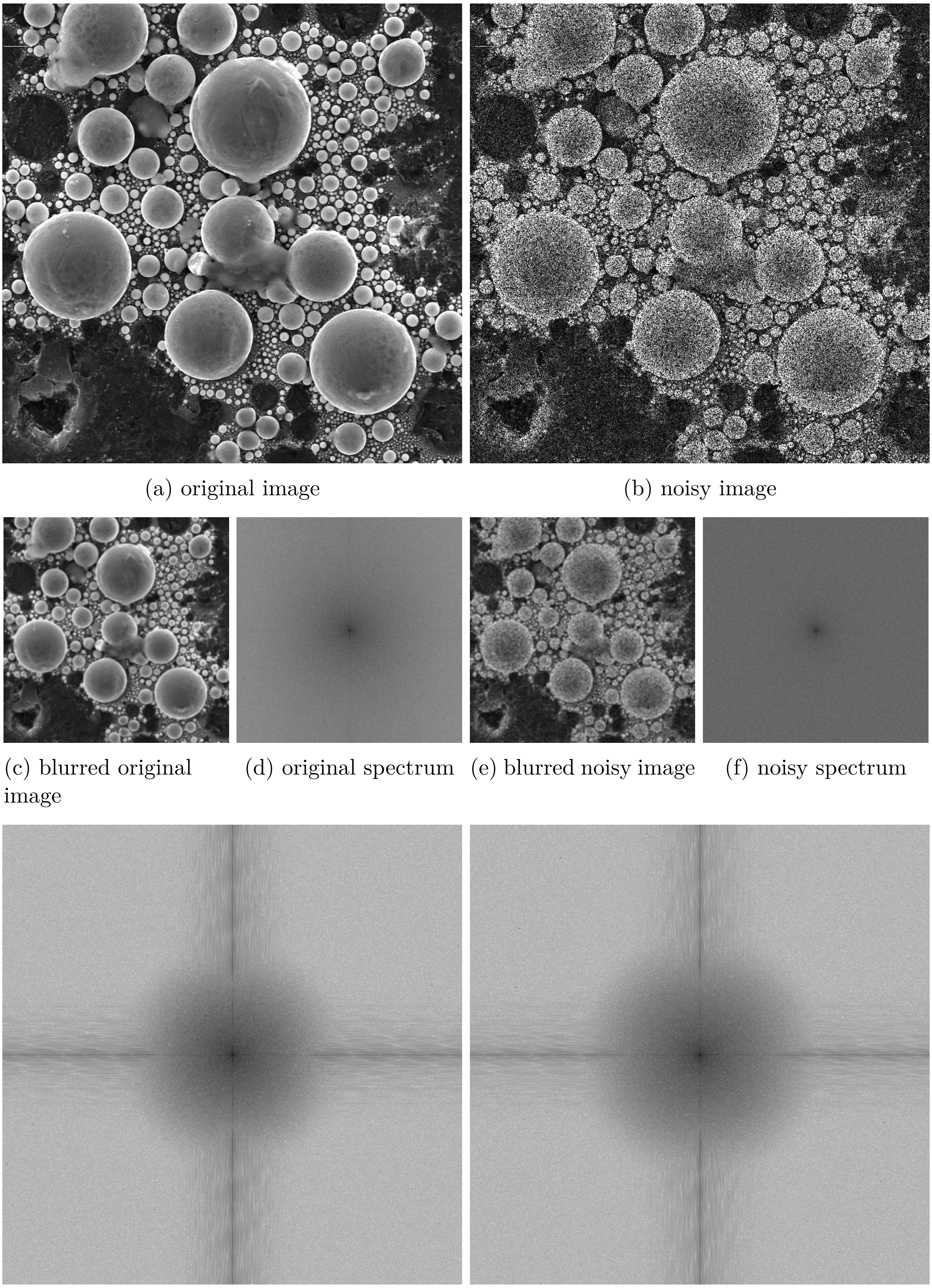
To compute the filtered image, a kernel $\boldsymbol{G}_\sigma$ is generated with values based on the weight function and normalized.[6][7] The filtered image is obtained by convolving the kernel with the original image. By taking the properties of Fourier transform into account, the filtered image can be computed as

$$\boldsymbol{W}_{\text{blur}} = \boldsymbol{W}_{\text{orig}} * \boldsymbol{G}_\sigma = \mathcal{F}^{-1}\{\mathcal{F}\{\boldsymbol{W}_{\text{orig}}\} \cdot \mathcal{F}\{\boldsymbol{G}_\sigma\}\}$$

Images in figure 5 show the effect of Gaussian blur. Applying a low-pass filter means that sudden changes in pixel values will be removed and the edges of structures blurred, the degree to which the image is affected depends on the size of the structures and the σ of the filter.

[6] The size of the kernel corresponds to the size of the image, in some implementations only a portion of the kernel is generated, for example 4σ, while the remaining elements are regarded as insignificant and truncated to zero to save computational time.

[7] Using a normalized kernel preserves brightness and contrast of the image.



(a) original image

(b) noisy image

(c) blurred original image

(d) original spectrum

(e) blurred noisy image

(f) noisy spectrum

(g) spectrum of the blurred original image

(h) spectrum of the blurred noisy image

Figure 5: Effects of using Gaussian blur ($\sigma = 2$) on an example image (SEM image of Sn particles on carbon). Noise was added to the image for the comparison. The plots of spectra use a logarithmic scale normalized to the original image, the ”fftshift” function was used to move the origin into the center of the image.

3.5 Windowing

Phase correlation provides a global maximum in the correlation matrix at coordinates (x_0, y_0). However, the Fourier transform periodizes the image, and because pixel values at opposing edges are generally different, a sharp transition corresponding to the edge will be observed in the resulting periodic function. As the edge can generally contain a part of a structure and therefore become a part of the spectrum with significant contrast, it is necessary to lower the impact of contrasting edges using a window. This is done by multiplying the image by a window function that tapers at the edges.[3] To leave most of the image unaffected by the window a tapered cosine window is used. The effect is shown in figure 7. The function

$$w_{\alpha,N}(k) = \begin{cases} \frac{1}{2} + \frac{1}{2}\cos\Big(\pi(\frac{2k}{\alpha(N-1)} - 1)\Big) & 0 \le k < n \\ 1 & n+1 \le k \le N-1-n, \\ \frac{1}{2} + \frac{1}{2}\cos\Big(\pi(\frac{2k}{\alpha(N-1)} - \frac{2}{\alpha} + 1)\Big) & N-1-n < k \le N-1 \end{cases}$$

where N is the number of samples and $n = \lfloor \alpha \frac{N-1}{2} \rfloor$, is called the one-dimensional tapered cosine window (also known as the Tukey window) with shape parameter α.[8] A rectangular window for two perpendicular dimensions can be obtained by multiplying two one-dimensional windows, one for each respective dimension.

$$w_{\alpha,N,M}(x,y) = w_{\alpha,N}(x) \cdot w_{\alpha,M}(y)$$

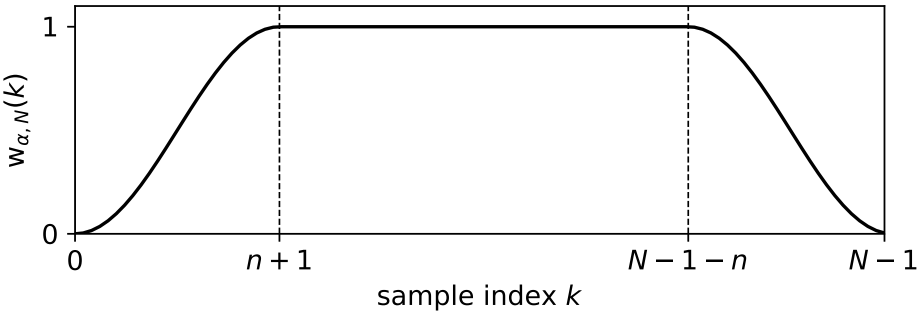
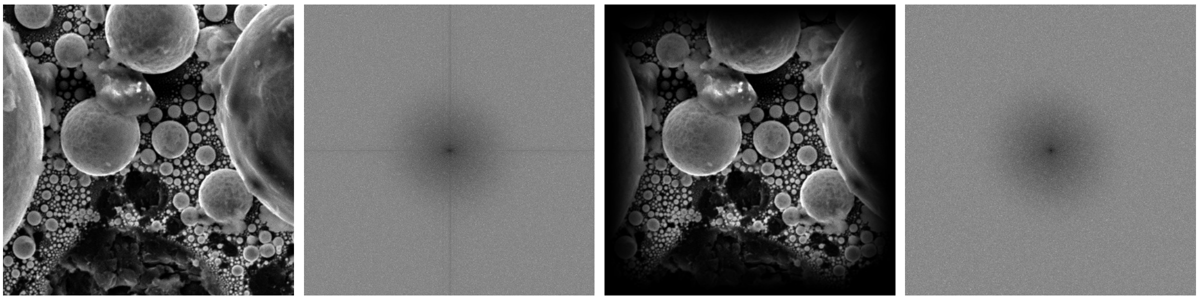


Figure 6: Graph of a one-dimensional tapered cosine window



(a) original image (b) original spectrum (c) after cosine window (d) spectrum after cosine window

Figure 7: Effect of cosine window ($\alpha = 0.5$) on the spectrum of an example image (SEM image of Sn particles on carbon). The plots of spectra use a logarithmic scale, the "fftshift" function was used to move the origin into the center of the image.

4 Drift compensation

The algorithms described below use image registration to track the motion of the objects visible by the SEM and moving the view field of the SEM using the beam shift functionality. The algorithm aims to provide good synchronization of the position of the view field with the object, while keeping the object stationary when no drift occurs.

Based on previous experience with temperature drift at the apparatus, it can be assumed that drift occurs much slower than scanning. As long as this assumption holds, two consecutive frames can be compared for drift.

Provided the change between consecutive frame is small, the change in the spectrum corresponding to the shift prevails over the development.

4.1 Feedback loop

First tests were performed with the simplest approach to take, which is a proportional feedback loop. A shift between two consecutive frames is detected. The previous change in the view-field position is subtracted and the view field is moved by the remaining distance.

Algorithm 1 Basic drift compensation

```
$(sx, sy) \leftarrow (sx_0, sy_0)$                          ▷ initial view field position
$(px, py) \leftarrow (0, 0)$                                ▷ previous shift
loop
    $image \leftarrow$ GetImage( )
    if $previous\_image$ is defined then
        $(dx, dy) \leftarrow$ ComputeShift($previous\_image, image$)
    else
        $(dx, dy) \leftarrow (0, 0)$
    end if
    $previous\_image \leftarrow image$
    $sx \leftarrow sx + (dx - px) \cdot units\_per\_pixel$
    $sy \leftarrow sy + (dy - py) \cdot units\_per\_pixel$
    $(px, py) \leftarrow (dx, dy)$
    DisplayImage($image$)
    MoveViewFieldTo($sx, sy$)
end loop
```

A limitation of the algorithm is that it relies on the accuracy of image registration. During testing of the algorithm it was discovered that it is prone to oscillations caused by rounding errors when a pixel-precise image registration is used. A major disadvantage is that the position of the SEM view field must be changed every frame unless the object is completely still.[8]

[8] The scanning coils take some time to stabilize after beam shift, which causes either a delay or jitter in the next frame.

4.2 Feedback loop with a buffer

A method inspired by a feature in cathode-ray-tube screens referred to as "overscan"[9] was used to surpass the need for setting the beam shift every frame. The view field of the SEM determining the scanned area and thus frame size is expanded to accommodate a buffer zone. Based on the desired size of the output image, a visible area is defined withing the frame relative to the center of the frame. The visible area moves relative to the center of the frame based on the shift detected between consecutive frames.

Algorithm 2 Drift compensation using overscan

```
$lx \leftarrow (view\_field\_N - visible\_area\_N)/2$          ▷ limits for the visible area
$ly \leftarrow (view\_field\_M - visible\_area\_M)/2$
$(sx, sy) \leftarrow (sx_0, sy_0)$          ▷ initial view field position
$(x, y) \leftarrow (0, 0)$          ▷ position relative to view field
loop
    $frame \leftarrow$ GetImage( )
    $image \leftarrow$ ExtractVisibleArea($frame, x, y$)
    if $previous\_image$ is defined then
        $(dx, dy) \leftarrow$ ComputeShift($previous\_image, image$)
    else
        $(dx, dy) = (0, 0)$
    end if
    $x \leftarrow x + dx$
    $y \leftarrow y + dy$
    $image \leftarrow$ ExtractVisibleArea($frame, x, y$)
    DisplayImage($image$)
    $previous\_image \leftarrow image$
    if $|x| \geq lx \vee |y| \geq ly$ then
        $sx \leftarrow sx + x \cdot units\_per\_pixel$
        $sy \leftarrow sy + y \cdot units\_per\_pixel$
        MoveViewFieldTo($sx, sy$)
        $(x, y) \leftarrow (0, 0)$
    end if
end loop
```

The output image is extracted from the frame based on the visible area. If a part of the visible area happens to be located outside of the frame due to a large shift, the part of the frame within the visible area is extracted and copied into a blank image which is the same size as the visible area. The shift of the subsequent frame is then detected relative to the previous visible area corrected for the previous sift.

The approach could not be relied on if the blank area played an important role in the spectrum of the image as it would majorly change the calculated shift. As seen in figure 9, the Fourier spectra of an image with a small section blank is so similar to the original image that no shift is detected.

[9] Overscan in a CRT causes a part of the image to be drawn outside of the visible area of the screen. More information can be found at https://en.wikipedia.org/wiki/Overscan

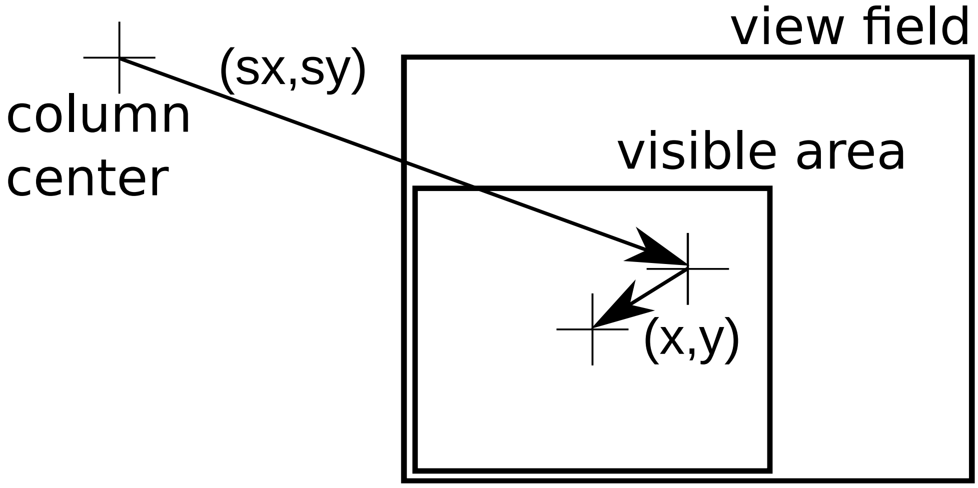
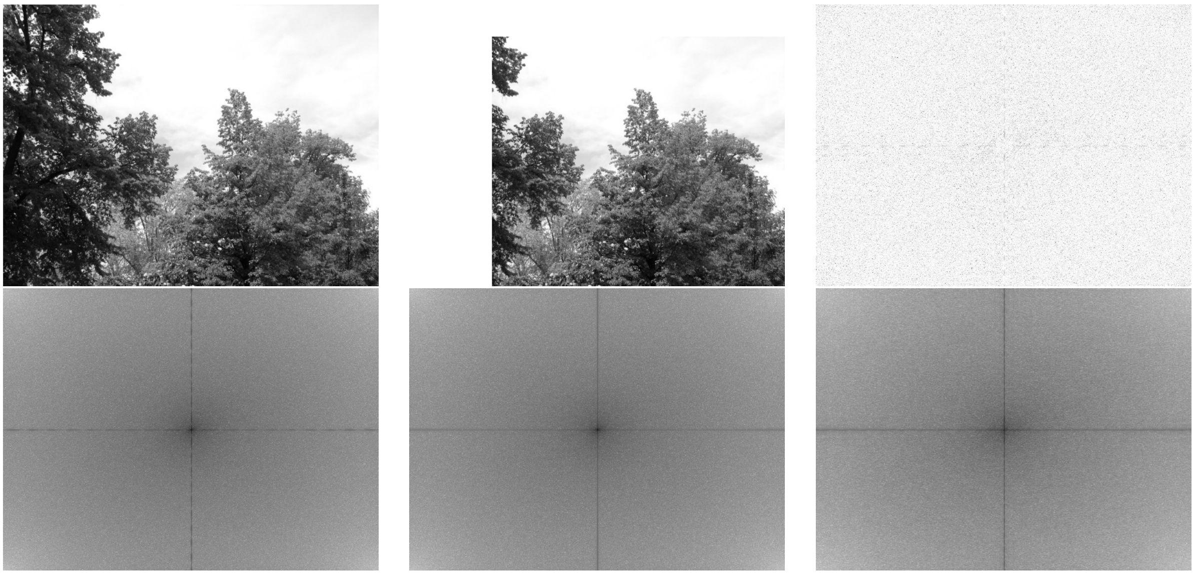


Figure 8: Illustration of overscan. (sx_0, sy_0) is the original view field position; (sx, sy) is the view field position; (x, y) is the position of the visible area within the frame; $(sx, sy) - (sx_0, sy_0) + (x, y)$ yields total drift for the frame.



(a) full image

(b) cropped image

(c) normalized ratio of the spectra (top), normalized cross-spectrum (bottom)

Figure 9: Spectral comparison between a cropped image and a full image. The spectrum plots use a logarithmic scale and were normalized, the ”fftshift” function was used to move the origin into the center of the image.

The main advantages are that frames can be corrected for drift in-situ and the SEM is spared from constantly changing the position of the view field. A big advantage is also that the object in the image can be stabilized even in cases where moving the view field is not possible (such as when processing a video). The only change affecting the result is that the pats of the image outside the view field are cropped.

5 Designing the application for drift compensation

The theory described in previous chapters intends to illustrate the operating principles of the experimental apparatus as well as outline the requirements for an application facilitating the compensation of drift for the SEM. This chapter describes the steps taken while designing the application and outlines the working principles behind it. Below are the most important requirements for the application

To begin with, it must be stated that the application is supposed to be a tool that helps further researchers as well as students working with the UHV experimental apparatus. It is therefore advisable to design a user-friendly application which does not require extensive cooperation. Reliability in conjunction with appropriate speed of operation must be taken into account.

The application must provide a real-time view of the current image acquired from the SEM, because the control software is unable to scan while remote scanning is in progress. Another requirement is the ability to record a video of the SEM images accompanied by the information about the drift of each frame. A plot tracing the trajectory of the object would be shown alongside the image, allowing the operator to observe the drift in real-time together with the image.

The application uses a multi-threaded model. The GUI runs on the main thread while scanning and recording are handled by dedicated threads. The computer on which the application will run features a recent multi-core processor, which should allow for the threads to run concurrently.

The application is divided into several objects to structure the functionality. Classes for the main window and widgets handle the GUI, while a separate class handles scanning and recording. A class is dedicated to motion tracking. The SEM library in particular was designed with future usability in mind.

A text-based JSON configuration file present in the application directory is used to set parameters like image size, overscan buffer size. A default file is created on startup if there is no configuration file present.

5.1 Implementation

Due to my previous experience with C++, it was a good candidate to implement this application as well, however, my advisor suggested to use Python due to the popularity it enjoys among other collaborators working on the design of the UHV apparatus. My experience with Python used to be very limited, which provided an interesting learning opportunity[10].

During testing in development, I discovered that the application requires graphical user interface (GUI) ; because I have previously used the Qt framework[16] to prepare a GUI, I was keen to discover the existence of the PyQt5 library[14], which enables straightforward use of Qt with Python.

[10] More information about python, including installation instructions, can be found on the website https://www.python.org/ and related sites.

5.1.1 Fourier transform complexity

Shift detection is the second (after scanning) most time-consuming process limiting frame rate. The discrete Fourier transform is a rather computationally intensive algorithm with the complexity of $\mathcal{O}(N\log(N))$ for a 1D case [10], therefore with the complexity of $\mathcal{O}(N \times M\log(N \times M))$ for a 2D case, because a 1D DFT must be computed for each row of the image matrix.

The FFT implementation included in the numpy library is, however, very inefficient, so the FFTW[5] library is used instead. To make a rough comparison of the speed of the FFT implementations, a test averaging shift detection on 1000 images sized 640 × 480 was performed on a testing machine with the following results

function	frame time [s]	input data type
numpy.fft.fft2	1	using 64-bit float (the only available)
scipy.fftpack.fft2	1	using 64-bit float
scipy.fftpack.fft2	0.7	using 32-bit float
FFTW	0.25	using 64-bit complex

The above comparison suggests that by using the FFTW library, the time required to compute a shift can be decreased to approximately one quarter.

5.1.2 Used libraries

The application was designed to work with the following libraries: numpy[12] v. 1.13.3, imageio[9] v. 2.2.0, pyFFTW[7] v. 0.10.4, FFTW[5] v. 3.3.7, Qt[16] v. 5.6.2, PyQt5[14] v. 5.6.0, Matplotlib[11] v. 2.2.0.

After encountering the need for an algorithm with sub-pixel resolution, I discovered a complex image registration library intended for astronomical images[6], which was briefly used during development until a custom solution was implemented.

5.2 Controlling the scanning electron microscope

The electron microscope control software supplied by Tescan provides a local TCP/IP interface for user applications, creating a server to which a client application can connect and send commands to the control software in order to change properties such as magnification, active detector, etc. as well as capture images – remote scanning. When a client is scanning remotely some settings are imported from the server, however others (such as scanning speed) must be set by the client. It is thus unnecessary (and perhaps would be counterproductive) to re-implement a user interface corresponding to the full functionality of the control software.

A binary protocol is used to communicate with the control software. Therefore, a library managing the communication was implemented using the protocol reference guide provided by Tescan. The library is split into two objects. The first is the ”SemConnection” class that handles the low-level network interface, building, sending, and receiving packets. All packet-related functions in the SemConnection class object are thread-safe. The second is the ”Sem” class that serves as a wrapper for the ”SemConnection” class. Each function (or command) described in the reference has a particular name, the input and output parameters are defined by order and data type. Because of this the command methods of the ”Sem” class just call a method of the ”SemConnection” class with arguments corresponding to the command: command name, expected output parameters, input parameters. Wrapper classes that pack and unpack a parameter exist for all data types supported by the protocol. The data type of each parameter can thus be defined by the corresponding wrapper class. To ease potential use of the library by future applications communicating with a Tescan electron microscope, all functions in the ”Sem” class were named in accordance with the protocol reference guide.

The application only uses a part of the functions in the protocol, which primarily consists of detector selection, magnification, scanning speed, size of the view field, and beam shift. Only the required commands are therefore included in the library, however, the remaining commands can be easily added by a future user if necessary.

5.3 Recording experiments

The application also implements the ability to record video of the experiment. The video is stored in the mp4 container encoded using the h.264 codec. The imageio[9] library is used to encode video.[11]

Recording the images from the SEM alone would not be sufficient, it is necessary to accompany the video with a file that stores metadata describing the trajectory of the object, which helps with interpreting the video. A human-readable format based on CSV was chosen. The first line starts with the ”#” (number sign) character followed by a dictionary of imaging parameters at the start of recording. The following lines contain data columns which are (in order): frame number (from 1), total X drift (in millimeters), total Y drift (in millimeters), seconds since the start of recording (seconds), changed parameters. The changed parameters column is an empty string unless a parameter changed, then it will be a dictionary of changed parameters. All the parameters are described in the following table.

Keys in the dictionary of parameters

key	datatype	unit	description
hv	float	V	SEM voltage
vf	float	mm	view field
wd	float	mm	working distance
spot	float	nm	spot size
dwell	float	µs	dwell time
recorded	datetime[12]		start of recording

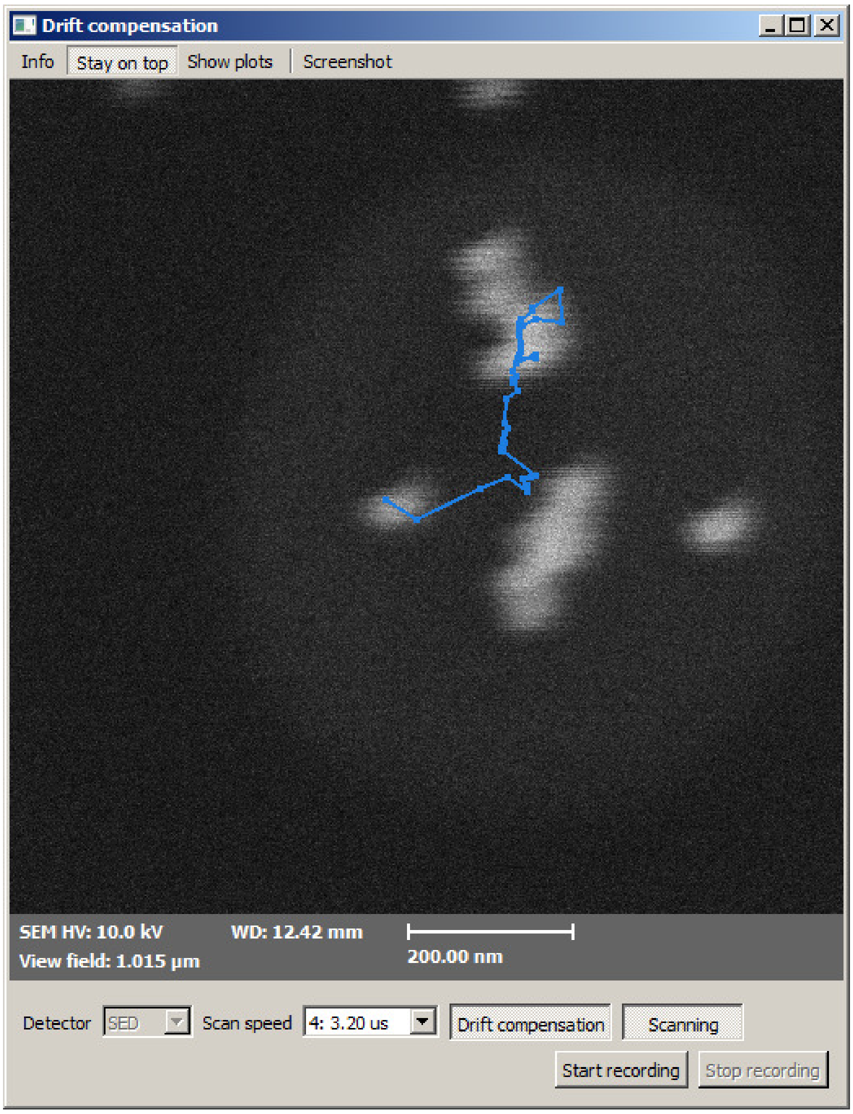
[11] imageio serves as a wrapper for FFMPEG (see https://www.ffmpeg.org/)

[12] as produced by the datetime.datetime.now function in Python

5.4 User interface

Throughout testing of the algorithm, it became clear that a console-based interface with a display window is insufficient, and a graphical user interface (GUI) is required for comfortable control.

Looking for the appropriate user interface, I decided to try out a multi-document interface (MDI) , which would contain sub-windows with widgets, however, subsequently I realized this approach heads in the direction of trying to re-implement parts of the control software, which I wanted to avoid.



SEM HV: 10.0 kV

WD: 12.42 mm

View field: 1.015 μm

Detector SED

Scan speed 4: 3.20 us

Figure 10: A screenshot of the main window with an example image

The GUI consists of a main window which displays the image from the SEM. A small control panel would be located below the image to set essential imaging parameters for the application. Plots of the drift is shown in a separate window opened on request. A menu[13] with additional controls is placed at the top of the window.

[13] A QToolBar had to be used to produce the menu, because setting a QMenu (or a top level QAction) in a QMenuBar checkable does not produce a checkable button.

5.4.1 Control and interaction

The user can chose the detector from which to capture, which is necessary as the order of detectors given by the SEM is not guaranteed. Scan speed can also be selected. The application selects the scan speed currently configured in the control software by default, while change in scan speed in the application has no effect on the scan speed selected in the control software. The ”Screenshot” button on the menu can be used to save the current image including the legend. The ”Info” button opens a menu to display information about the SEM device and version, application, and the Qt toolkit. The ”Stay on top” button asks the window manager to keep the window on top of the stack when toggled.

Toggle buttons are used to turn scanning and drift compensation on or off. When scanning is initiated, the first frame is not compared for drift, which allows the user to observe drift, then disable scanning to make changes in the control software, and enable scanning to observe drift once again. Drift compensation can be turned on or off at any time. The selected detector cannot be changed while scanning is on. Toggling scanning off does not stop recording, the recorder will just remain waiting until a new image is available, however, closing the application stops recording gracefully.

5.4.2 Image widget

The images scanned from the SEM are passed to a widget by the scanning thread using a thread-safe function once a frame has been scanned and processed. Each image is passed with a corresponding shift relative to the last image in pixels, which is used to draw a series of lines tracing out the trajectory of the object over the image.

A legend showing the image parameters is drawn under the image. The legend is drawn once (using QPainter) and cached until a parameter changes. The legend features a scale bar, which adjust size depending on the size of the view field using the function in algorithm 3.

Algorithm 3 Calculate the scale bar length, both in pixels l and view field units d

```
Require: maximum length in pixels $l_{\max} \geq 0$, units per pixel $px > 0$
  function ScaleBarLength($px$, $l_{\max}$)
      $d \leftarrow px \cdot l_{\max}$                      ▷ estimate the length in units $d$
      $d \leftarrow \text{ROUND}(d, -\lfloor \log_{10} |d| \rfloor)$    ▷ round $d$ to one significant figure
      $l \leftarrow \text{ROUND}(d/px)$                     ▷ actual length in pixels
      if $l > l_{\max}$ then
          $l \leftarrow l/2$
          $d \leftarrow d/2$
      end if
      return ($l$, $d$)
  end function
```

5.4.3 Plot window

The plot window is shown in figure 11. To show the drift of the object in a clear manner, it is displayed separately for each axis. The data is updated together with the image, however, the widget re-draws the plot only when it is visible. The main purpose is to provide a way to monitor an experiment, while the metadata file can be used for later processing. Matplotlib[11] is used to draw the plots as it has a Qt-compatible output – to embed a figure in a Qt UI, Matplotlib provides a Qt backend matplotlib.backends.backend_qt5agg.FigureCanvas, that produces a Qt widget to which the figure is drawn. The position of the object is passed to the widget using the Qt event system. The scale on the axes is changed automatically based on the data currently displayed. A button to erase all the values from the plot and a button to redraw the plot are present in the widget.

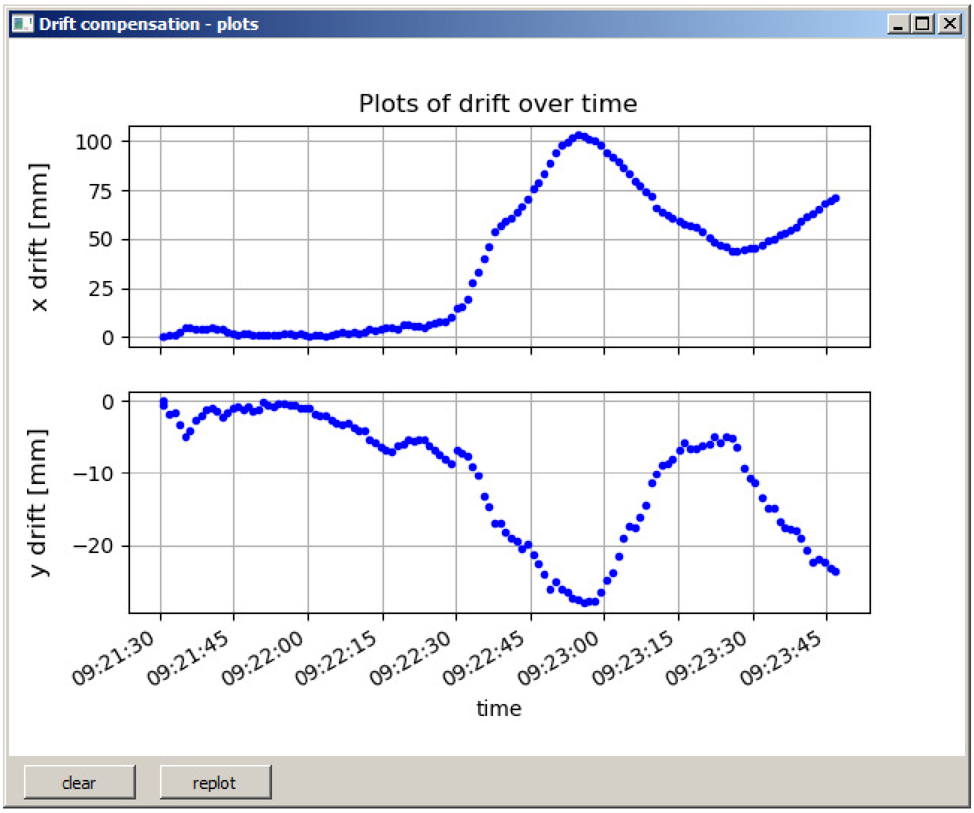


Figure 11: A screenshot of the plot window with example data

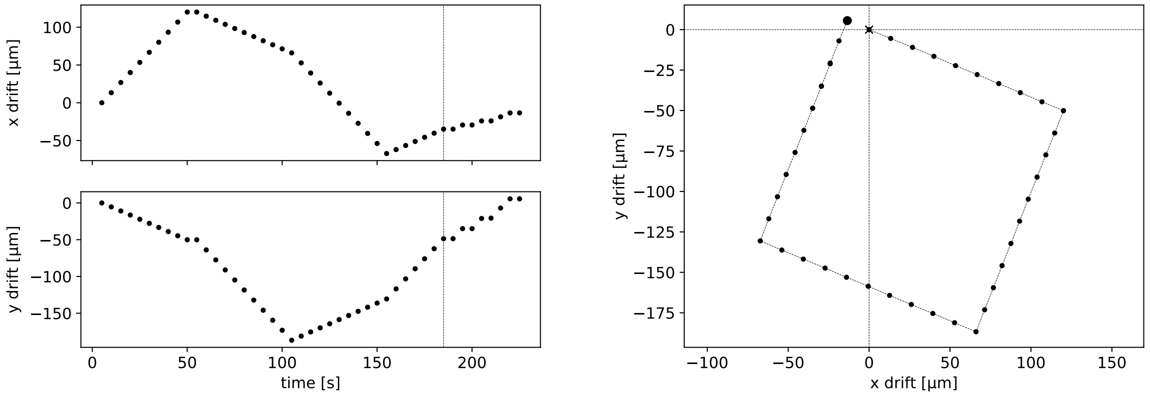
6 Experimental verification of the proposed solution

After implementing the application according to the design, it had to be tested in practice to ensure proper functionality. Because of ongoing development of the apparatus, a special setup was required to install a sample heater. The fist test was therefore performed using a manipulator as the cause of drift. After the application was found to function as intended, an experiment with the heated holder was performed.

6.1 Testing using a manipulator

To practically verify the functionality of the application before an experiment with a heated sample, the algorithm was tested using a manually controlled manipulator to move the sample. The application does not have the ability to communicate with the manipulator, which means that it could not possibly take the motion into account and thus perceives it as drift.

Before the experiment it was necessary to set the period of scanning to a value that ensures the manipulator has stabilized after a step, which prevents jitter in the scanned image. For the installed manipulator this was about 3 s for steps of 10 µm. A calibration sample featuring a rectangular grid of gold particles on carbon was used. After enabling drift compensation, the manipulator was used to trace out a square 110 µm × 110 µm in size. For most of the experiment, I was trying to keep the speed at which the manipulator is advanced constant at one step per frame, however, for the last several steps (transition marked by a line in figure 12 (a)) the speed was intentionally decreased to two frames per step to make sure no drift is detected when the object remains at rest. As noticeable from the plots in figure 12, the axes of the manipulator are not aligned with the scanning axes. The plots show that the algorithm can not only detect shifts between images, but it can also be used to measure the distance traveled by an object.



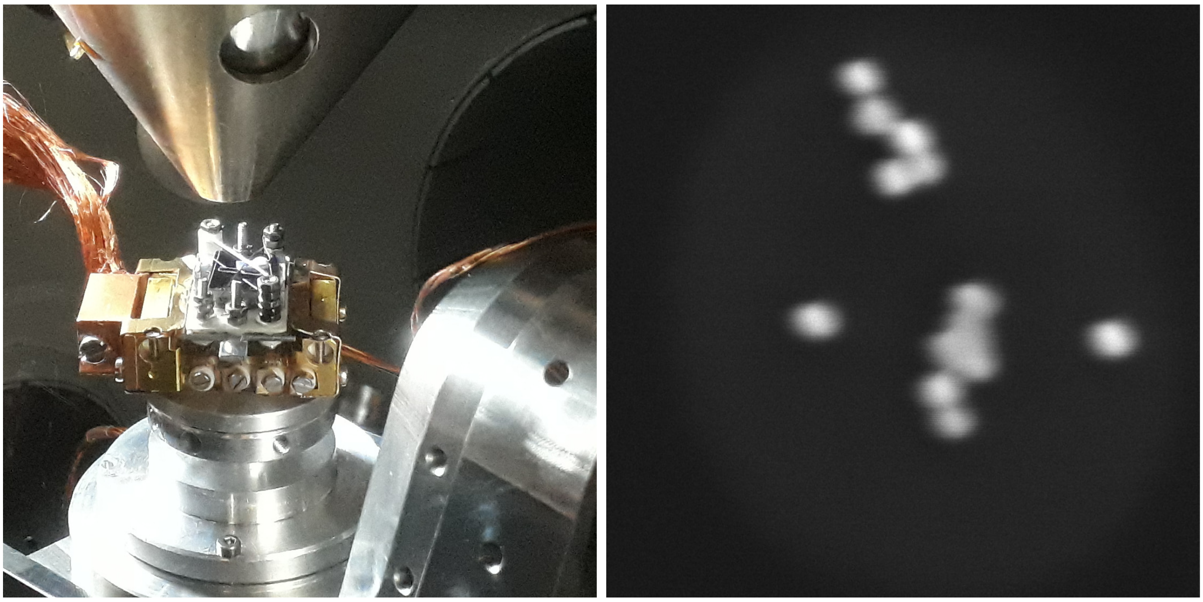
(a) plot of drift in the x and y axes

(b) plot of the trajectory of the object created by combining the X and Y axes

Figure 12: Testing drift detection using a manipulator

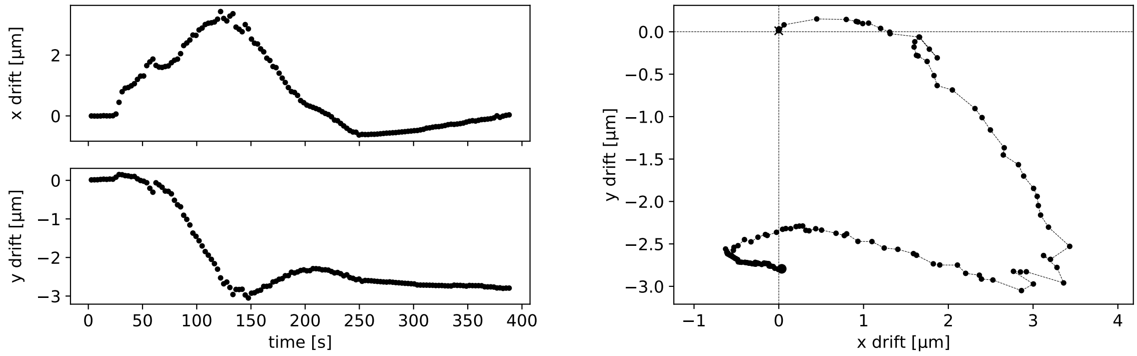
6.2 Examining temperature drift

After the functionality of the application was verified using a manipulator, the next step is to heat up a sample and observe the temperature drift. A silicon wafer with an SiO_2 on which are gold particles was used. The heater consisted of a resistive silicon wafer mounted on a palette, the sample placed atop the wafer. A calibrated thermometer was not available for the heater at the time of the experiment, however the peak power was about 1.5 W. A suitable formation of particles was found on the sample at a magnification of approximately 136 kx, which was done to ensure proper functionality at high magnification close to the limit of the imaging capability of the SEM. The experiment consisted of two parts, first the sample was heated by gradually increasing the power. After about 140 seconds, the power is gradually decreased until the heater is turned off.



(a) the holder with the sample in the chamber (b) SEM image of the structure tracked

Figure 13: image of the sample heated during the experiment



(a) plot of drift in the x and y axes (b) plot of the trajectory of the object created by combining the X and Y axes

Figure 14: Plot of the drift observed on the heated sample

The observed region of the sample returns to the original position in the X direction, but stays approximately at the position to which it has drifted in the Y direction. This could be caused by the fact that the sample was not firmly attached to the heater.

7 Conclusion

My thesis examines the issue of temperature drift encountered during the use of an SEM to observe an object in a wide range of temperatures. I devised an algorithm that uses motion tracking based on Fourier transform to detect and compensate for drift. Based on the input from my advisor, I designed an application that seamlessly integrates with the current control software used for the SEM, which uses the motion tracking algorithm to change the position of the view field on to keep the originally viewed area of the sample in the view field. The application can also record detected temperature drift and video from the SEM images during an experiment.

I experimentally verified that the algorithm functions as intended and tested the application as it would be used by future users.

Due to current technical limitations of the apparatus it was not possible for the application to implement manipulator communication, the availability of which would extend the possibility to correct drift beyond about 200 µm the SEM beam shift can handle. It would also be possible to compensate drifting faster than scanning speed. If the object is lost, the veiw field would be extended (effectively zooming-out), the object would be found in the larger image, a manipulator move the sample, and finally the view field would be zoomed back in.

The application could also be extended to correct the working distance based on the change in focus between images.

Although capable of standalone function, the algorithm and application that were presented could perhaps be taken just as a proof-of-concept the essence of which later becomes a part of a more complex control software that could be able to manage all (or at least most of) the functionality required to operate the whole UHV experimental apparatus.

Future use of drift compensation in nanostructure analysis might not be limited to keeping an object in the view field. Change in length due to a difference in temperature could be inferred from the drift, which could be used to determine the coefficient of thermal expansion corresponding to a sample. Vice-versa, the knowledge of the thermal expansion coefficient could be used to measure a change in temperature based on the difference in length.

References

[1] ARGYRIOU, V., VLACHOS, T. *A study of sub-pixel motion estimation using phase correlation.* [online] [cit. 2018-04-10]. Available from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.4780&rep=rep1&type=pdf

[2] AXMAN, Tomáš. *Vývoj nosiče vzorků pro měření elektrických vlastností v UHV SEM.* Brno University of Technology. Faculty of Mechanical Engineering, 2016. (40 pages). Advisor Ing. Petr Bábor, Ph.D.

[3] DRUCKMÜLLEROVÁ, Hana. *Phase-correlation based image registration*, Brno: Brno University of Technology, Faculty of Mechanical Engineering, 2010. (100 pages). Advisor Mgr. Jana Procházková, Ph.D.

[4] DRUCKMÜLLER, Miroslav. *Phase correlation method for the alignment of total solar eclipse images.* The Astrophysical Journal, 2009, vol. 706, no. 2, pp. 1605–1608. ISSN 0004-637X.

[5] FRIGO, Matheo, JOHNSON, Steven G. *FFTW.* [online] 2018 [cit. 2018-04-10]. Available from http://www.fftw.org/

[6] GINSBURG, Aadam. *(Astronomical) Image Registration* [online] 2017 [cit. 2018-04-10]. Available from http://image-registration.readthedocs.io/en/latest/

[7] GOMERSALL, Henry. *pyFFTW 0.10.4 documentation.* [online] 2016 [cit. 2018-04-10]. Available from https://hgomersall.github.io/pyFFTW/

[8] HARRIS, Fredric J. *On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform.* 1978. Proceedings of the IEEE 66 (1): 51-83. DOI:10.1109/PROC.1978.10837

[9] KLEIN, Almar. *imageio - Python library for reading and writing image data.* [online] 2018 [cit. 2018-04-10]. Available from https://imageio.github.io/

[10] LOHNE, Mathias. *The Computational Complexity of the Fast Fourier Transform.* [online] 2017 [cit. 2018-04-10]. Available from: https://folk.uio.no/mathialo/texts/fftcomplexity.pdf

[11] The Matplotlib development team. *Matplotlib: Python plotting.* [online] 2018 [cit. 2018-04-10]. Available from https://matplotlib.org/

[12] Numpy developers. *NumPy.* [online] 2018 [cit. 2018-04-10]. Available from http://www.numpy.org/

[13] REIMER, Ludwig. *Scanning electron microscopy: physics of image formation and microanalysis.* 2nd edition. Berlin: Springer, 1998. ISBN 978-3-642-08372-3.

[14] Riverbank Computing Limited. *PyQt5 Reference Guide.* [online] 2017 [cit. 2018-04-10]. Available from http://pyqt.sourceforge.net/Docs/PyQt5/

[15] SHAPIRO, Linda. STOCKMAN, George. *Computer vision.* pages 153-156. Prentice Hall, 2001.

[16] The Qt Company Ltd. *Qt 5.6.* [online] 2017 [cit. 2018-04-10]. Available from http://doc.qt.io/archives/qt-5.6/index.html

Symbols

$\mathbb{N}$	the set of natural numbers
$\mathbb{R}$	the set of real numbers
$\mathbb{R}_0^+$	the set of non-negative real numbers
$\mathbb{C}$	the set of complex numbers
(x, y)	vector (x, y)
$\boldsymbol{A}(x, y)$	the element of $\boldsymbol{A}$ at position (x, y)
$\boldsymbol{A} \cdot \boldsymbol{B}$	the element-wise multiplication of $\boldsymbol{A}$ and $\boldsymbol{B}$
$\frac{\partial \boldsymbol{A}}{\partial x}$	the derivative of $\boldsymbol{A}$ in the direction of x
$\boldsymbol{W} * \boldsymbol{G}$	the convolution of $\boldsymbol{W}$ and $\boldsymbol{G}$
a^*	the complex conjugate of a
$\lfloor a \rfloor$	the floor function, the largest integer less than or equal to a
$\|a\|$	the absolute value of a
$\mathcal{F}$	the (discrete) Fourier transform (this work does not use the continuous FT)
$\mathcal{F}^{-1}$	the (discrete) inverse Fourier transform
$\arg\max\{\boldsymbol{A}\}$	the coordinates of the maximum of $\boldsymbol{A}$

Abbreviations

CSV	coma-separated values
DFT	discrete Fourier transform
FT	Fourier transform
GUI	graphical user interface
JSON	JavaScript Object Notation
MDI	multi-document interface
SEM	scanning electron microscope
TCP/IP	Transmission Control Protocol/Internet Protocol
UHV	ultra high vacuum